

**Пояснювальна записка
до дипломного проекту
на тему:
«Платформа для автоматизації учбового процесу»**

Київ – 2019 рік

ЗМІСТ

ВСТУП	2
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	4
2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ.....	9
2.1 WordPress	9
2.2 Joomla	18
2.3 Drupal.....	22
3 АНАЛІЗ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ	28
3.1 PHP	28
3.2 PostgreSQL	30
3.3 Фреймворк	34
3.4 Composer	39
4 ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЕКТУ	42
4.1 Проектування схеми бази даних	42
4.2 Маршрутизація	46
4.3 Контролери	47
4.4 Представлення.....	51
5 ПРОЦЕС ДЕПЛОЮ ТА ПРИКЛАД ВИКОРИСТАННЯ.....	53
5.1 Деплой	53
ВИСНОВКИ.....	59
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60

					IT41.100БАК.005 ПЗ			
Зм	Лист	№ докум.	Підп.		Платформа для автоматизації учбового процесу Пояснювальна записка	Літ.	Аркуш	Архивів
Розроб.	Єпанча						1	60
Перев.	Катін							
Н.контр.						КПІ ім.Ігоря Сікорського		
Затв.						Група IT-41		

ВСТУП

З розвитком комп'ютерних систем, почалася глобальна автоматизація процесів. Ми знаходимося на такому етапі автоматизації, що більша частина процесів вже автоматизованому. Однак ще залишилися процеси, для яких автоматизація не була реалізована або була, але не в повному обсязі. Однією з таких областей є освіта.

У нашій країні існує не так багато закладів, у яких введено хоча б часткову автоматизацію процесів. Але і це працює не на рівні Міністерства освіти, а виключно як експерименти на базі певного закладу. З реалізованих вже процесів ми можемо розглядати такі процеси як:

- сторінка викладача закладу;
- журнал оцінок;
- щоденник;
- відомості відвідування.

Для імплементації цих систем в учбовий процес використовуються різні засоби, які, здебільшого, потребують розвернення локального сервера. Але не у кожного учбового засобу є така можливість. Сам це і є ключовим фактором, що сповільнює розвиток автоматизації.

Саме тому необхідно сконцентрувати увагу на створенні засобу, що вирішує цю проблему. Одним з можливих рішень, є створення хмарної платформи, що буде дозволяти підключити заклад у декілька дії і не потребує додаткових навичок для його використання. Головною особливістю цього проекту може стати, ще це єдина платформа для різних типів закладів:

- шкіл;
- вищих навчальних закладів;
- приватних закладів.

Це допоможе людям використовувати уніфіковану платформу для будь-якого типу навчання без зв'язку з певним навчальним закладом. Першим етапом

					IT41.100БАК.005 ПЗ	Лист
Зм	Арк	№ докум.	Підпис	Дата		2

для реалізації такої платформи стане реалізація системи для викладачів. Буде розроблено систему, що дозволить розміщувати інформацію про викладача та певні його матеріали.

У цій роботі буде докладно проаналізовано структури освіти в Україні як предметну область. Далі, на основі результатів досліджуваної області, буде сформовано докладну задачу для створення платформи. Сама платформа має складатися з модулів, кожен з яких слугую для певної цілі. Розробка модульної системи зараз є актуальною оскільки саме такий підхід надає змогу розширювати систему без втручання у інші частини. Це допоможе у майбутньому побудувати систему, що буде стійка до втручання та модифікацій.

Слід зазначити, що данній роботі буде розроблено один із модулів для платформи, як приклад розробки частини модульної системи. Сам модуль буде слугувати для розміщення матеріалів викладачів, а також, можливо, для створення сторінок для студентів або персонала. Отже, система не бути мати обмеження для використання по ролі користувачів.

					IT41.100БАК.005 ПЗ	Лист
Зм	Арк	№ докум.	Підпис	Дата		3

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Освіта — цілеспрямована пізнавальна діяльність людей з отримання знань, умінь та навичок або щодо їх вдосконалення. Процес і результат засвоєння особистістю певної системи наукових знань, практичних умінь та навичок і пов'язаного з ними того чи іншого рівня розвитку її розумово-пізнавальної і творчої діяльності, а також морально-естетичної культури, які у своїй сукупності визначають соціальне обличчя та індивідуальну своєрідність цієї особистості.[1]

Також дисципліна, яка вивчає переважно методи навчання і вивчення в школах або подібних закладах у протилежність різним неформальним засобам соціалізації (наприклад, між батьками і їх дітьми). Тому освіту можна розуміти як передачу накопичених суспільством знань молодому поколінню для розвитку в нього пізнавальних можливостей, а також набуття умінь і навичок для практичного застосування загальноосвітніх і професійних знань.

Освіта призначена, щоб дати новому поколінню початкові знання культури, формуючи поведінку у дорослому житті і допомагаючи у виборі можливої ролі в суспільстві. У широкому сенсі слова, освіта — процес або продукт "...формування розуму, характеру або фізичних здібностей особистості... У технічному сенсі утворення — це процес, за допомогою якого суспільство через школи, коледжі, університети та інші інститути цілеспрямовано передає свою культурну спадщину — накопичене знання, цінності та навички — від одного покоління іншому «між поколіннями». В останні десятиліття в країнах Заходу набула широкого розповсюдження наукова експертиза у сфері освіти, яка здійснюється у формах «педагогічного аудиту», «психолого-педагогічної експертизи», «комплексної гуманітарної експертизи» тощо.[1]

Для коректної постановки задачі по створенню платформи для автоматизації учбового процесу необхідно у першу чергу проаналізувати актуальну структуру освіти у країні[2].

					ІТ41.100БАК.005 ПЗ	Лист
Зм	Арк	№ докум.	Підпис	Дата		4

Українська структура формується Законом України «Про освіту» и включає у себе:

- дошкільна освіта;
- загальну середню освіту;
- позашкільна освіта;
- професійно-технічна освіта;
- вища освіта;
- післядипломна освіта;
- аспірантуру;
- докторантуру;
- самоосвіта.

Дошкільна освіта для українців, як і для багатьох європейців, це освіта, що починається з дошкільного навчального закладу, зазвичай - з дитячого садка. У 2 (а іноді з 1,5) року малюк потрапляє в ясла, з 3 - в молодшу групу. За бажанням батьків дитина може залишатися в саду півдня, а в разі проблем зі здоров'ям - і цілодобово. Для запису в дитсадок існують онлайн-черги, де можна вибрати заклад і записати в нього дитину відразу після народження.

Іноді сім'я приймає рішення навчати дитину вдома, але з 2001 року здобуття дошкільної освіти є обов'язковим для дітей 5-річного віку. Батьки можуть вибрати дитсадки або групи короткотривалого перебування, зараз в Україні діє понад 1000 таких груп. При загальноосвітніх навчальних закладах працюють групи підготовки дітей до навчання в школі. Майбутні школярі отримують там базові знання, а також знайомляться зі шкільним укладом, вчителями і однокласниками. З 6 років дитина може відправлятися в школу.

Загальна середня освіта є обов'язковими і виходить в різних типах навчальних закладів, переважно - в середній загальноосвітній школі. Існують три ступені, які представлені нижче:

I - початкова школа (1-4 класи), надає початкова загальна освіта;

II - основна школа (5-9 класи), надає базову загальну середню освіту;

					IT41.100БАК.005 ПЗ	Лист
						5
Зм	Арк	№ докум.	Підпис	Дата		

III - старша школа (10-11 класи), надає повну загальну середню освіту.

Завдання шкільного навчання - всебічний розвиток дитини, надання йому тієї бази знань, яку вимагає від неї суспільство. У школі повинні сприяти професійному самовизначенню, фізичному розвитку і формування основних морально-етичних норм.

Обдаровані діти можуть надходити в профільні класи з поглибленим вивченням окремих предметів або початкової допрофесійної підготовки, спеціалізовані школи, гімназії та ліцеї. Особливо талановитим дозволено претендувати на державну підтримку - стипендії, направлення в провідні вітчизняні чи зарубіжні освітні центри. Можна взагалі не відвідувати школу - учень має право прискорено здобути середню освіту і здати іспити екстерном.

Додаткове розвиток здібностей і талантів досягається у позашкільній освіті. До таких навчальних закладів належать спортивні секції, школи мистецтва, різноманітні учнівські клуби і подібні організації, державні або приватні.

Професійно-технічна освіта орієнтується на отримання певної спеціальності. Отже, за ним слід відправлятися в ті освітні установи, які навчають робітничим професіям. Це професійні училища та ліцеї (художні, технічні, вищі), училища-агрофірми і заводи, навчально-курсіві комбінати та подібні заклади. До них же відносяться центри професійної освіти, підвищення кваліфікації та перепідготовки робітничих кадрів.

Професійно-технічні навчальні заклади часто співпрацюють з підприємствами - замовниками підготовки співробітників. Для отримання робочої професії учень взагалі не зобов'язаний відвідувати навчальний заклад, адже пройти підготовку можна безпосередньо на виробництві.

Структура вищої освіти України побудована відповідно до освітніх системами розвинених країн світу, які курируються ЮНЕСКО, ООН та іншими міжнародними організаціями. Вона забезпечує фундаментальну наукову, професійну та практичну підготовку, перепідготовку та підвищення кваліфікації студентів.

					IT41.100БАК.005 ПЗ	Лист 6
Зм	Арк	№ докум.	Підпис	Дата		

Форми навчання у вищих навчальних закладах різноманітні: денна, вечірня, заочна, дистанційна або комбінована, яка об'єднує кілька з цих форм.

Для вищих навчальних закладів наведено нижче існуючі чотири рівні акредитації:

I - технікум, училище та прирівняні до них вищі навчальні заклади;

II - коледж та прирівняні до нього вищі навчальні заклади;

III і IV (залежно від результатів акредитації) - інститут, консерваторія, академія, університет.

З 2008 року обов'язковою умовою вступу до вищого навчального закладу є проходження зовнішнього незалежного оцінювання (ЗНО). У 2019 році учні будуть здавати тести з таких предметів:

- українська мова та література;
- історія України (з найдавніших часів до наших днів);
- хімія;
- фізика;
- біологія;
- математика;
- географія;
- англійська, іспанська, німецька, російська і французька мови.

Після здобуття вищої освіти можна отримати науковий ступінь - в аспірантурі і докторантурі. Це передбачає написання і захист дисертації - кандидатської чи докторської відповідно - а також придбання наукового та викладацького досвіду.

Післядипломна освіта дає можливість отримати нову кваліфікацію, спеціальність чи професію. Найчастіше це відбувається на основі вже отриманої освіти в попередньому навчальному закладі, хоча можна отримати і зовсім нову спеціальність. До закладів післядипломної освіти належать академії, інститути (центри) підвищення кваліфікації або перепідготовки, відповідні підрозділи вищих навчальних закладів, професійно-технічні навчальні заклади, науково-

					ІТ41.100БАК.005 ПЗ	Лист
						7
Зм	Арк	№ докум.	Підпис	Дата		

методичні центри професійно-технічної освіти та підрозділи в організаціях та на підприємствах.

Для самоосвіти можливості надзвичайно широкі - можна відвідувати лекторії, наукові центри і різні громадські організації. Найбільш перспективний напрямок самоосвіти пов'язано з мережею Інтернет. Популярними стають онлайн-курси, які дозволяють дивитися відеолекції відомих фахівців в різних наукових сферах, перевіряти свої знання за допомогою тестування, спілкуватися з іншими студентами і викладачами. Сьогодні це один з найбільш популярних векторів освіти в Європі та Америці, а в Україні скористатися подібними можливостями можна на навчальній платформі дистанційної освіти EDUGET.

Проаналізувавши систему освіти в Україні, можна зробити висновок, що система буде актуальною для використання для закладів середньої освіти, вищої освіти, дистанційної, а також післядипломної освіти. Однак програмне забезпечення може бути використаним і для приватної освіти, оскільки підхід до навчання не відрізняється.

У даній роботі вибрано реалізації одного модулю для організації учбового процесу, а саме мікроблог для викладачів. У цьому блозі кожен викладач зможе керувати своєю сторінкою. Йому буде надано можливість розміщувати інформацію про себе, свій досвід, освіту, також ділитися своїми публікаціями.

					IT41.100БАК.005 ПЗ	Лист
						8
Зм	Арк	№ докум.	Підпис	Дата		

2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

Для огляду існуючих рішень будуть використовуватися CMS оскільки саме вони будуть максимально прості та зручні для вирішення конкретної задачі по створення мікроблогу для викладачів. Використання модульної системи для конкретної задачі є оптимальним підходом, оскільки кожен модуль стає ізольованим від інших, що дає можливість уникнути проблем при редагування самих модулів.

Для порівняння були вибрано три найпопулярніші CMS системи. Перш ніж ознайомитися з тим, як ці CMS відрізняються один від одного, вам також будуть приведені деякі довідкові дані та статистику. Таким чином, у вас буде краще уявлення про ці рішення.

2.1 WordPress

WordPress (вимовляється «Вордпрес») — система керування вмістом з відкритим кодом, яка через свою простоту в установленні та використанні широко застосовується для створення веб-сайтів. Сфера використання — від блогів до складних веб-сайтів. Вбудована система тем і плагінів у поєднанні з вдалою архітектурою дозволяє конструювати на основі WordPress практично будь-які веб-проекти.[3]

WordPress вперше був випущений в 2003 році. Його основна технологія - PHP з базою даних SQL у фоновому режимі. В останні роки, як і багато інших сайтів в Інтернеті, JavaScript стає все більш важливим. Це особливо очевидно з таких проектів, як Calypso і майбутній редактор Gutenberg.

З моменту свого створення, WordPress безперечно мав найкраще виконання всіх доступних систем управління контентом.

На ринку CMS, це велика риба в ставку з ринковою часткою 60 відсотків і ця цифра зростає. Крім того, він скоро буде працювати на третину всього Інтернету. WordPress нещодавно перетнув 30-відсотковий поріг загального

					IT41.100БАК.005 ПЗ	Лист 9
Зм	Арк	№ докум.	Підпис	Дата		

використання. За даними Live Stats близько 560 мільйонів веб-сайтів, запущено на WordPress. Це також робить WordPress найшвидше зростаючою CMS за останні вісім років.

Тільки останню версію (4.9) було завантажено понад 90 мільйонів разів. Серед його користувачів є деякі з найбільш відомих сайтів. Від Нью-Йорк Таймс до TechCrunch та Forbes - багато великих імен там вибирають WordPress для керування своїм сайтом.

Важливим фактором для створення веб-сайту є те, скільки це коштуватиме. Що спільне у всіх трьох системах керування контентом у нашому порівнянні CMS - це те, що вони є відкритими кодами та повністю вільними для використання. Майте на увазі, що створення веб-сайту досі приносить витрати, навіть якщо CMS безкоштовна.

Вам потрібно заплатити за хостинг. Залежно від того, який тип хостингу ви використовуєте, ціни будуть відрізнятися, але хостинг досі є найдешевшим рішенням. Вам також знадобиться доменне ім'я, який приносить додаткові витрати. Це стосується всіх трьох.

Найбільш вирогідними додатковими витратами для WordPress є преміальні плагіни, теми та розширення. Хоча CMS пропонує безліч безкоштовних матеріалів, для деяких речей все ще необхідно і має сенс витрачати гроші. Це особливо вірно, якщо ви працюєте на професійному сайті.

Наприклад, WooCommerce, найпопулярніше рішення для електронної комерції, є безкоштовним. Однак розширення, такі як нові платіжні шлюзи та інші функції, потрібно купувати на офіційному ринку. Ціни коливаються від 750 до 7500 грн.. Цей тип бізнес-моделі і ціноутворення є досить типовим для сфери WordPress. Є також купа безкоштовних розширень.

Іншим фактором вартості може бути професійна допомога, яка може знадобитися для реалізації більш складних речей на вашому сайті. Завдяки популярності WordPress такі послуги є доступними та часто використовуються.

Погодинна оплата на таких сайтах, як Upwork, сильно відрізняється, але ви легко знайдете когось у межах бюджету.

В цілому, WordPress є відмінним варіантом для створення веб-сайту за дешевою ціною. Майте на увазі, що платформа прагне бути дещо більшою, ніж інші два кандидати. З цієї причини це може збільшити вартість хостингу, коли ваш сайт зросте.

Зручність користування є однією з основних точок продажу систем управління контентом в цілому. Вони дали змогу менш технічно підготовленим людям створити сайт без навичок кодування. З цієї причини зручність використання є важливим фактором прийняття рішень, особливо для початківців. Таким чином, він повинен бути частиною будь-якого порівняння CMS.

Простота використання - одна з найбільших сил WordPress і одна з головних причин її успіху. Налаштування відбувається швидко і легко, і називається 5-хвилинною установкою не без причин.

Крім того, багато хостингових компаній (наприклад, Bluehost) пропонують опції інсталяції в один клік WordPress. Це робить створення веб-сайту не складнішим, ніж подання онлайн-форми.

Інтерфейс користувача WordPress дуже простий. Він пропонує менше можливостей, ніж ваш канал новин Facebook, і ви можете досягти більшості речей за допомогою простих клацань миші, наприклад, оновлення сайту. Останнє також дуже важливо для безпеки WordPress. Приклад панелі керування або адміністративної панелі наведено на Рисунку 2.1.

Створення вмісту також надзвичайно просте. Якщо ви можете використовувати звичайний текстовий процесор, ви можете створювати повідомлення та сторінки за допомогою WordPress. Процес інтуїтивно зрозумілий, включаючи додавання зображень та інших медіа до ваших повідомлень. WordPress також дозволяє вставляти вміст з багатьох зовнішніх служб за допомогою простого копіювання та вставки.

Для більш складних макетів є конструктори сторінок. Вони дозволяють початківцям створювати та змінювати вигляд сторінок за допомогою графічного інтерфейсу, а не кодування. Все це також дозволяє легко навчати WordPress клієнтам. Приклад редагування шаблону можна побачити на Рисунку 2.1.

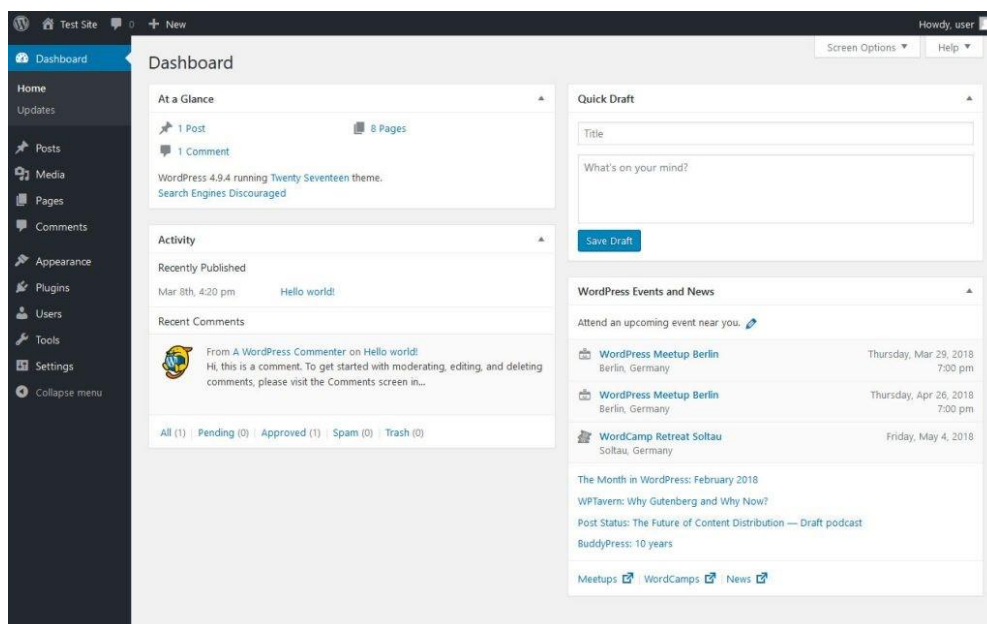


Рисунок 2.1 – Головна сторінка адміністративної панелі WordPress

Одна річ, яку платні рішення мають як перевагу для себе, це те, що вони пропонують визначену і центральну точку контакту. Якщо щось не працює, ви знаєте, з ким розмовляти. Рішення з відкритим кодом, які за визначенням не мають жодного власника, цього не мають. З цієї причини важливо, щоб вони ввели інші структури підтримки.

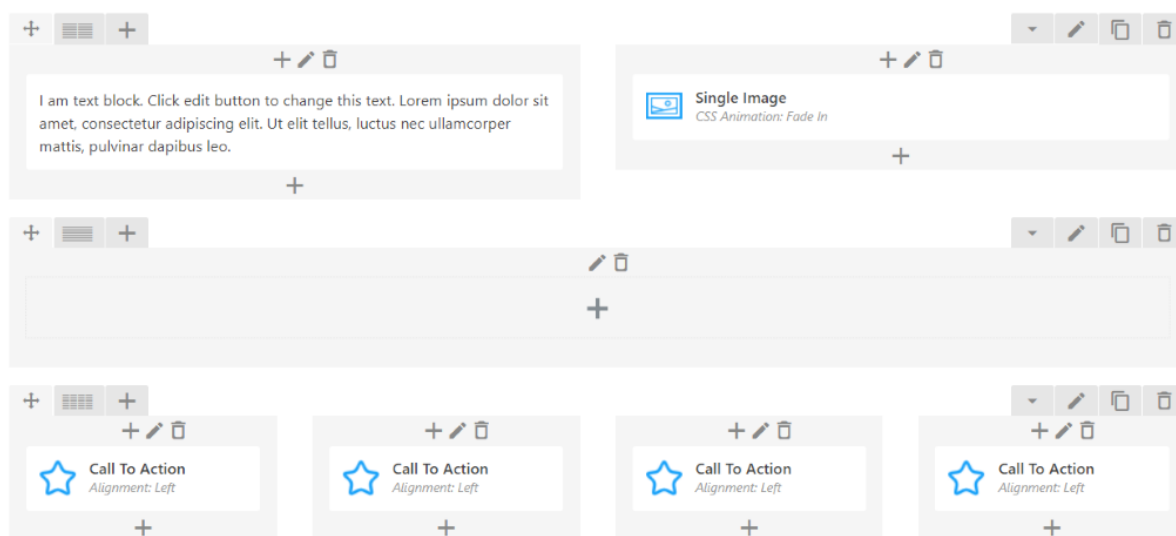


Рисунок 2.2 – Інтерфейс конструктора сторінок у WordPress

Однією з переваг використання WordPress є те, що ви станете частиною одного з найбільших інтернет-спільнот, що існують. Отже, існують великі можливості для підтримки. Особливо корисна стане документація, сторінка для якої наведено на Рисунку 2.3.

Офіційні форуми підтримки часто відвідують волонтери, які хочуть безкоштовно допомогти. CMS також пропонує безліч документів у формі WordPress Codex, документації для розробників і довідників.

Існує також широка мережа блогів, опублікованих на цю тему, так що ви можете знайти рішення ваших проблем, розміщених в інтернеті. Вищезгадані незалежні розробники та агентства пропонують платну підтримку, коли це необхідно.

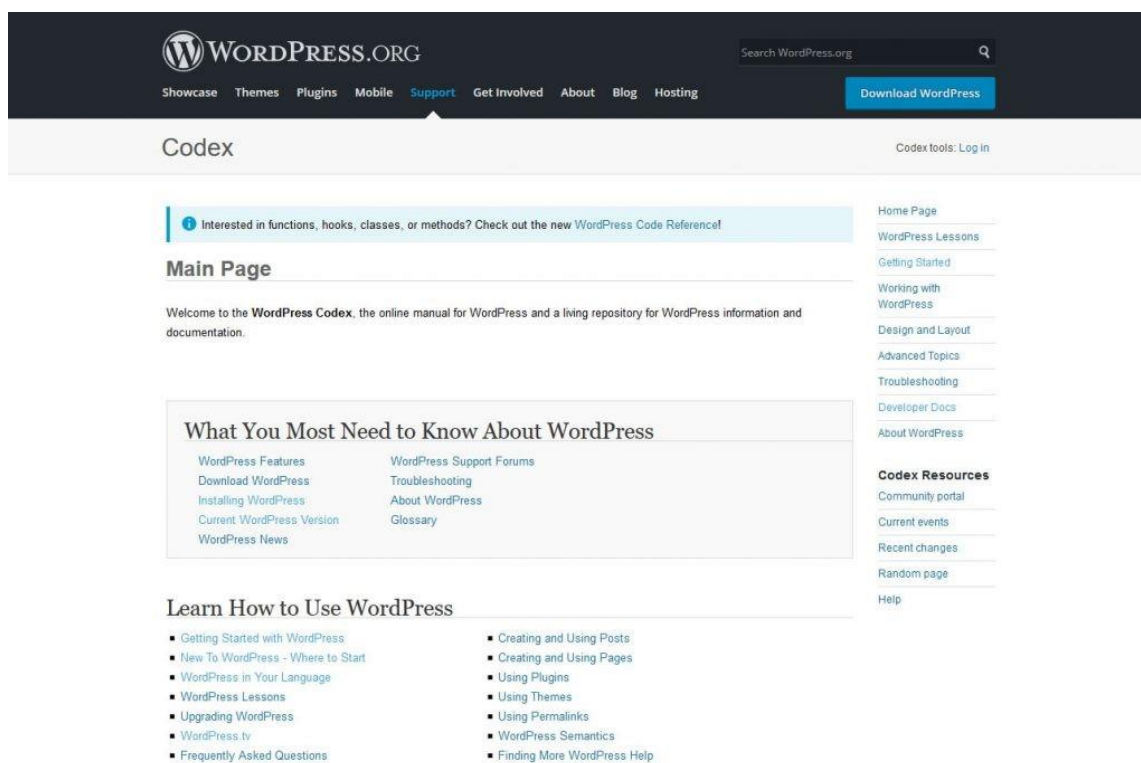


Рисунок 2.3 – Сторінка з документацією для WordPress

Крім того, у також є можливість піти на керований хостинг WordPress. Коли ви це зробите, ваш хост подбає про всі технічні частини вашого сайту. Таким чином, ви зможете повністю зосередитися на створенні контенту та маркетингу вашого сайту.

Сама платформа також добре підтримується, регулярні оновлення виходять кожні три-чотири місяці. Вони приносять нові можливості, поліпшення і заходи безпеки, щоб зберегти ваш сайт добре налагодженою машиною. Переконалися частоті оновлень можна за допомогою Рисунку 2.4.

May 17, 2018	WordPress 4.9.6 Privacy and Maintenance Release
April 3, 2018	WordPress 4.9.5 Security and Maintenance Release
February 6, 2018	WordPress 4.9.4 Maintenance Release
February 5, 2018	WordPress 4.9.3 Maintenance Release
January 16, 2018	WordPress 4.9.2 Security and Maintenance Release
November 29, 2017	WordPress 4.9.1 Security and Maintenance Release
November 16, 2017	WordPress 4.9 "Tipton"

Рисунок 2.4 – Список оновлень для WordPress

Можливість налаштування веб-сайтів, побудованих за допомогою CMS (і самої CMS), має першорядне значення для користувачів. Системи обслуговують дуже різноманітну базу користувачів і повинні вміти враховувати багато різних випадків. Тому, крім твердого продукту ядра, вони повинні запропонувати способи його розширення та модифікації.

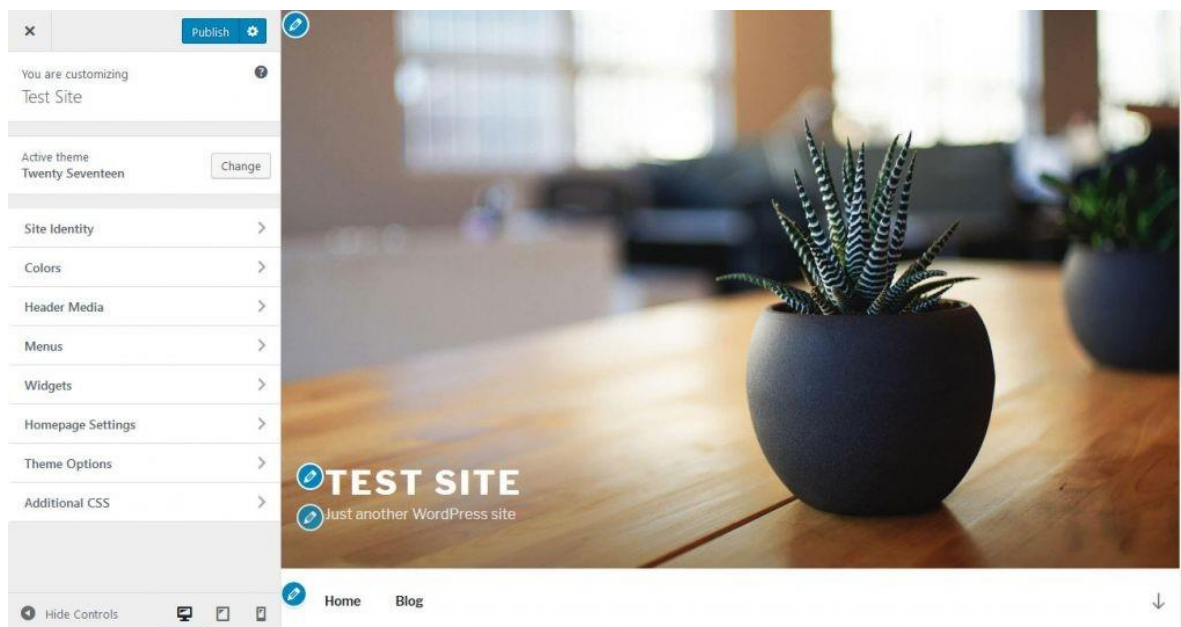


Рисунок 2.5 – Редагування шаблону у WordPress

Можливість налаштування - ще одна перевага платформи WordPress. Ви майже нічого не можете змінити. Навіть вбудовані опції дозволяють змінити

дизайн та функціональність. Від схеми кольорів адміністратора та дизайну сайту до спеціальних меню, віджетів, фонових зображень і зображень заголовка. Сторінку редагування шаблону сайту представлено на Рисунку 2.5.

Крім того, існує майже 55 000 плагінів і кілька тисяч тем, які тільки чекають розширення вашого сайту. Якою б метою чи функціями не вирішили доповнити сайт, ймовірно, вже існують спеціалізовані теми та / або плагіни для неї. Це навіть не стосується численних преміальних пропозицій. Сторінку для завантаження додаткових плагінів представлено на Рисунку 2.6.

Дочірні теми дають змогу змінювати майже все на своєму сайті безпечним способом. Таким чином, ви можете повністю зробити це відповідно до ваших потреб.

Якщо коротко, то WordPress може бути сформований в будь-який тип веб-сайту, який ви хочете.

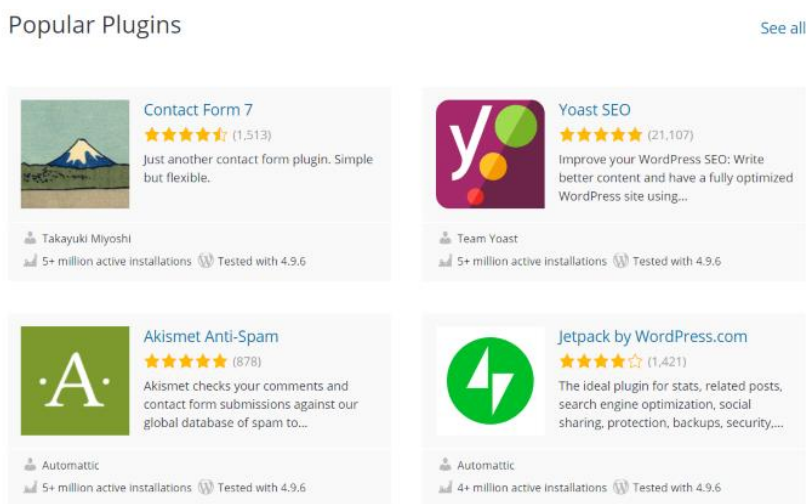


Рисунок 2.6 – Завантаження плагінів для WordPress

Час завантаження сторінок має значення як для відвідувачів, так і для пошукових систем. Ваш сайт може мати відзначений нагородами дизайн і найцінніший вміст, але якщо він не завантажується протягом декількох секунд, деякі люди ніколи не чекають його побачити.

Особливо це стосується мобільних пристроїв, які обігнали настільні комп'ютери в кількості використання. З цієї та інших причин продуктивність повинна бути важливою частиною будь-якого порівняння CMS.

Продуктивність часто називається одним з слабких місць WordPress. Це правда, що його зосередження на забезпеченні зворотної сумісності та підтримці старих версій трохи відстають у цьому відношенні. Водночас це частково через популярність платформи.

Хоча WordPress може бути найменш розширюваним з трьох CMS у цьому списку, він все ще може керувати великомасштабними веб-сайтами з часом завантаження сторінки до другої секунди. Але у той же час, можна побачити статистику використання, що наведена на Рисунку 2.7.

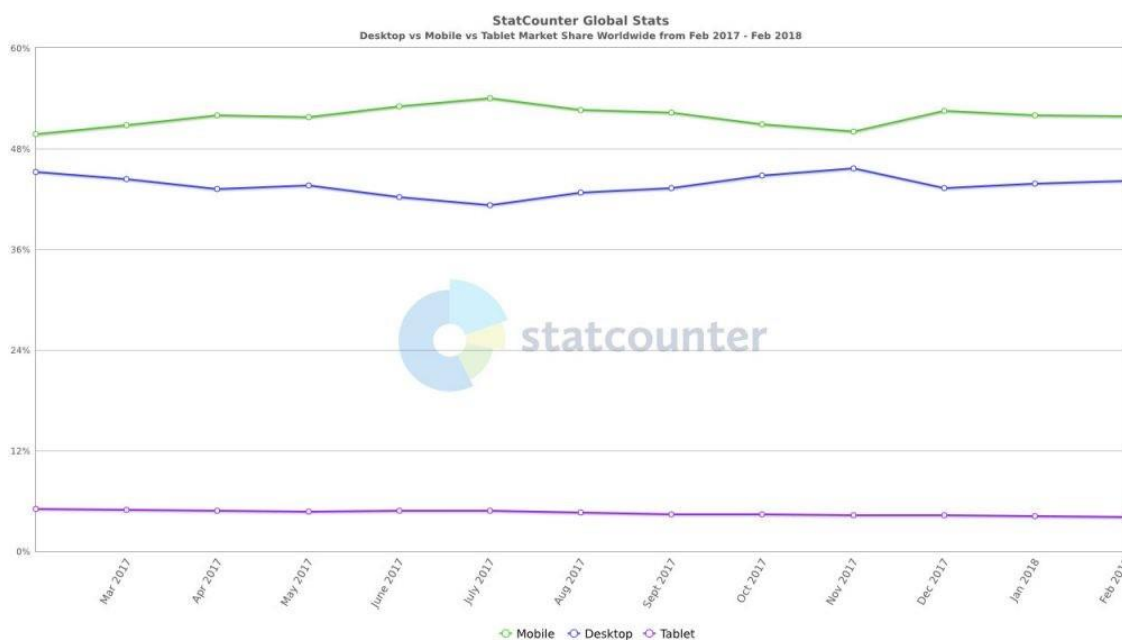


Рисунок 2.7 – Кількість відвідувань з різних платформ

Перш за все, є вищезгаданий керований хостинг WordPress. Якщо ви виберете його, ваш провайдер подбає про важку роботу з сайту, наприклад, кешування та навіть оновлення сайту. Це додає правильне обслуговування сайту, що вже дасть швидке завантаження сайту. Крім того, є ще багато речей, які ви можете зробити, щоб прискорити WordPress.

Платформа WordPress також показала, що вона може успішно обробляти сайти з тисячами сторінок і мільйонами щомісячних відвідувачів.

Будь-який власник вебсайту усвідомлює ризик, пов'язаний із запуском веб-сайту. Ті, хто не усвідомлюють ризик швидко розуміють його. Спам і спроби

автоматичного взлому - це вже велика проблема. Хоча велика частина тягаря за безпеку лежить на хостинг-провайдері, обрана CMS також повинна мати можливість боротися з нею.

У минулій версії WordPress отримав багато поганої відкликів, коли мова йде про безпеку. Деякі з них є небезпідставними.

Однак перспектива також трохи перекошена. Завдяки своїй популярності, WordPress також пропонує більшу ціль для хакерів. З такою широкою базою користувачів, є ще багато можливостей, щоб спробувати свою удачу.

Окрім цього, дотепер WordPress діє разом. Одним з прикладів є те, що всі сайти WordPress, що працюють у версії 3.7 і вище, тепер автоматично застосовують оновлення та технічне обслуговування. Тому основний продукт WordPress ніколи не стає незахищеним. Поточні проблеми безпеки найчастіше стосуються плагінів сторонніх виробників, а не самої CMS.

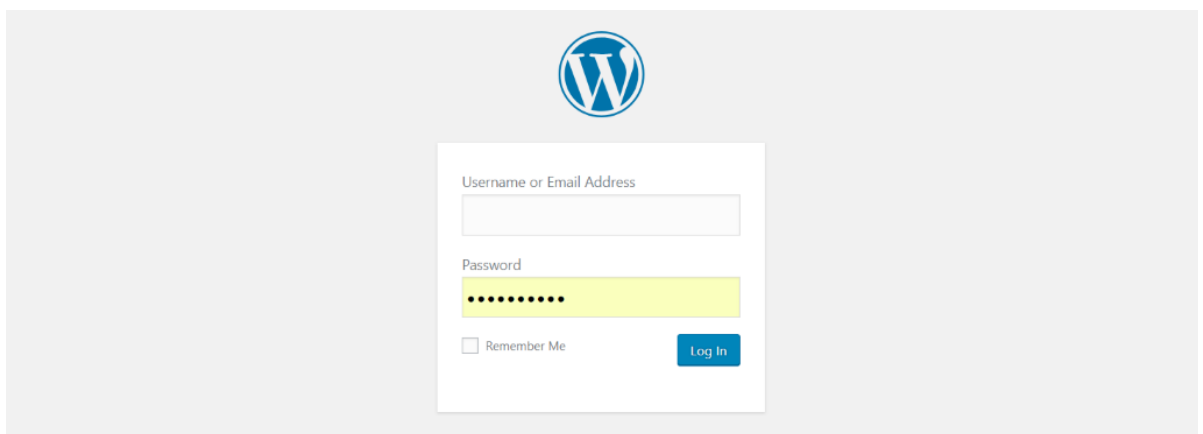


Рисунок 2.8 – Сторінка авторизації WordPress

У WordPress є вбудована система користувачів, що є необхідна у теперешній час на будь-якому сайті.

Крім того, одним з найбільших слабких місць для безпеки WordPress є користувач. Небезпечна інформація для входу та вебсайти, які не були оновлені, є одними з головних причин того, що сайти будуть зламані. З цієї причини ваша відповідальність - виховувати себе (наприклад, як перенести WordPress на HTTPS) і робити найкраще з вашого боку.

2.2 Joomla

Joomla! (вимова: «Джумла») — відкрита універсальна система керування вмістом для публікації інформації в інтернеті. Підходить для створення маленьких і великих корпоративних сайтів, інтернет порталів, онлайн-магазинів, сайтів спільнот і персональних сторінок. З особливостей Joomla можна відзначити: гнучкі інструменти управління обліковими записами, інтерфейс для управління медіа-файлами, підтримка створення багатомовних варіантів сторінок, система управління рекламними кампаніями, адресна книга користувачів, голосування, вбудований пошук, функції категоризації посилань і обліку кліків, WYSIWYG-редактор, система шаблонів, підтримка меню, управління новинними потоками, XML-RPC API для інтеграції з іншими системами, підтримка кешування сторінок і великий набір готових доповнень.[4]

Joomla! було створено в 2005 році, як вилка іншої CMS під назвою Mambo. Назва походить від Jumla, слово на суахілі, що означає "всі разом". Це стосується його природи як проекту з відкритим кодом, що підтримується спільнотою волонтерів.

Joomla є другим за популярністю CMS в Інтернеті. Його ринкова частка на ринку CMS це 6,3 відсотка, вона забезпечує 3,1 відсотка всіх веб-сайтів в Інтернеті. Це означає приблизно 58 мільйонів веб-сайтів. Загалом, він був завантажений більше 93 мільйонів разів.

Як і WordPress, Joomla написана на PHP і базі даних SQL. Інші характеристики також подібні, такі як поділ на інтерфейс (видима частина веб-сайту) і back-end (область адміністрування). Ви дізнаєтеся більше подібності нижче.

Витрати на роботу сайту Joomla схожі на WordPress. Це програмне забезпечення є безкоштовним, але вам, мабуть, доведеться приймати преміальні шаблони та / або розширення. Ціни в офіційному каталозі мають приблизно такий же діапазон, що і для WordPress.

					IT41.100БАК.005 ПЗ	Лист 18
Зм	Арк	№ докум.	Підпис	Дата		

Оскільки Joomla трохи складніше, ніж WordPress, ймовірність того, що вам буде потрібна професійна допомога, вище. Завдяки ринковим силам (менше постачання, підвищенню попиту), це може бути трохи складніше, але це все одно можна виконати. Ціновий діапазон для робіт з розробки приблизно однаковий у порівнянні з WordPress.

З точки зору складності, Joomla знаходиться десь між WordPress і Drupal. Якщо ви - людина з середньою технічною кваліфікацією, ви маєте можливість швидко потрапити до нього.

Вручну встановлення Joomla також схоже на WordPress. Просто завантажте програмне забезпечення, завантажте його на сервер і запустіть скрипт інсталяції. Багато хостинг-провайдерів також пропонують установку Joomla одним натисканням.

Крім того, CMS також надає графічний інтерфейс для додавання та керування статтями, носіями, меню, розширеннями та зміненням налаштувань.

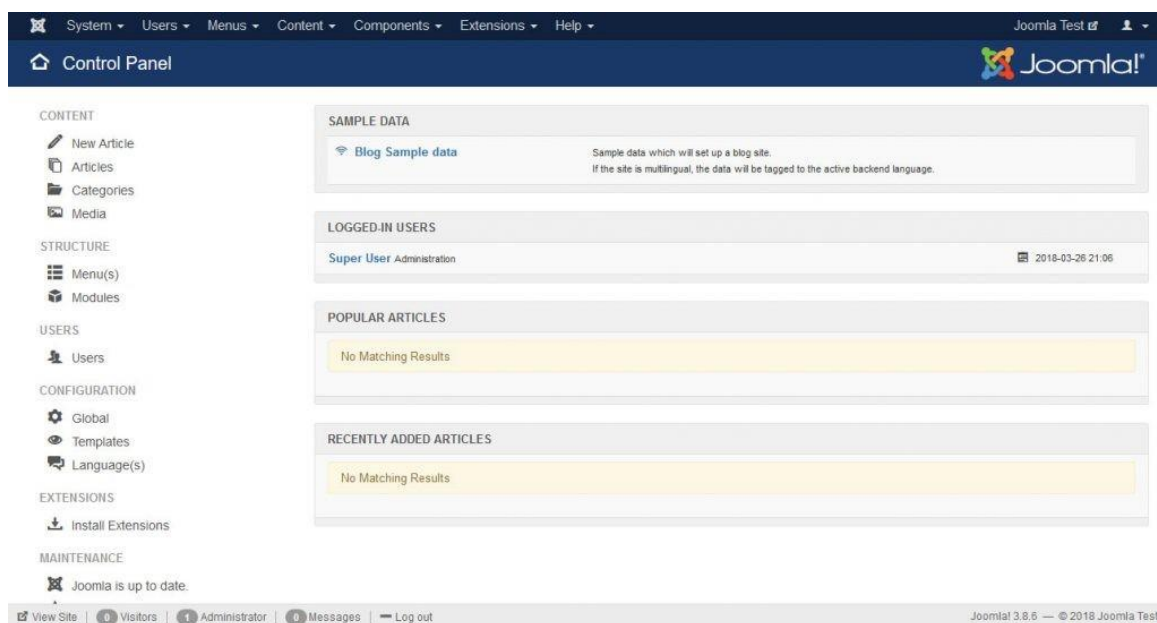


Рисунок 2.9 – Сторінка конфігурації Joomla

Тут також з'являються конструктори сторінок, які допомагають початківцям реалізовувати більш складні макети без коду. Joomla також поставляється з оновленням в один клік. В цілому, CMS пропонує більше функціональності з коробки, ніж WordPress, але, отже, також є більш складним. Також Joomla має

безліч налаштувань. На Рисунку 2.9 наведено приклад сторінки конфігурації, але слід зазначити, що вона може бути значно більшою.

Як друга за величиною CMS з відкритим вихідним кодом в Інтернеті, Joomla також має хорошу структуру підтримки. Є офіційним порталом допомоги. Ви можете задати питання і отримати технічну підтримку. Спільнота не така велика, як для WordPress, але ви все одно можете отримати відповіді швидко і легко. На Рисунку 2.10 зображено сторінку з документацією, що допоможе швидше почати роботу.

Крім того, ви можете копатися в документації, підписуватись на списки розсилки і навіть користуватися спеціальним чатом IRC. Подібно до WordPress, існують ресурси третьої сторони, такі як публікації в блогах і оплачені навчання онлайн. Також доступні професійні послуги від розробників.



Рисунок 2.10 – Офіційна документація Joomla

Що стосується технічного обслуговування, Joomla оновлюється частіше, ніж WordPress. Загалом, є нова мінорна версія кожні один-три місяці. Основні випуски виходять лише приблизно раз на рік.

Joomla також має тематичну та плагінну екосистему для додавання нових функцій на ваш сайт. Вони також мають багато інших типів розширень. На Рисунку 2.11 представлено сторінку для завантаження сторонніх додатків.

Шаблони - такі ж, як і для тем WordPress. Шаблони обробляють зовнішній вигляд, оформлення та розташування вашого веб-сайту, включаючи задню частину.

					IT41.100БАК.005 ПЗ	Лист 20
Зм	Арк	№ докум.	Підпис	Дата		

Компоненти - змінювати частини тіла сторінки та мати власні пункти меню в задній частині Joomla.

Модулі - еквівалент віджетів. Вони додають пошукові форми, хлібні крошки та інші компоненти сторінки.

Плагіни - це обробники подій, які виконуються лише за певних умов, наприклад, для блокування частин вмісту.

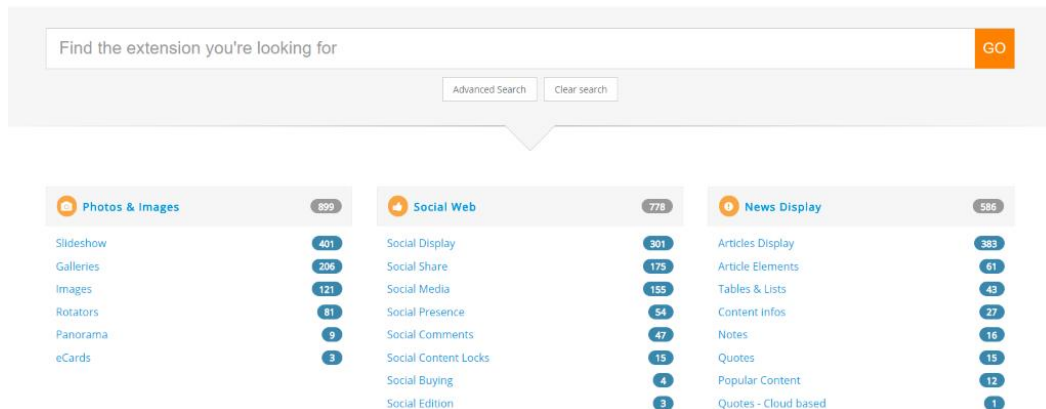


Рисунок 2.11 – Додаткові розширення для Joomla

У Joomla різниця в тому, що немає центрального каталогу для тем. Замість цього користувачам необхідно знайти власні довірені джерела.

Для компонентів, модулів і плагінів можна використовувати офіційний каталог. На момент написання цієї статті він містить майже 8000 розширень Joomla, доступних для пошуку за типом, категорією, тегами, сумісністю та багато іншого.

На відміну від WordPress, не все в каталозі безкоштовне. Хороший шматок складається з платних розширень. Крім того, не всі компоненти сумісні з усіма версіями Joomla. Тим не менш, те ж саме можна сказати про плагіни WordPress, які більше не підтримуються.

Joomla має хорошу репутацію щодо продуктивності. Однією з причин є вбудована функція підвищення продуктивності. Наприклад, можливо ввімкнути кешування та стиснення Gzip з адміністративної панелі. Він також поставляється з плагінами, щоб зробити його ще швидше.

Окрім цього, є необхідно прийняти звичайні заходи, як мінімізація плагінів і HTTP-запитів, оптимізація зображень тощо, щоб швидко завантажувати ваш сайт.

Існує також певний хостинг для Joomla, але це не так часто, як для WordPress. Приклад списку, що є необхідним для забезпечення безпеки сайту наведено на Рисунку 2.12.



Рисунок 2.12 – Список кроків для забезпечення безпеки сайту Joomla

Joomla також покладає більшу частину зусиль на те, щоб сайт був безпечним для користувача. Незважаючи на те, що співтовариство реагує на уразливості та створює латки, застосування безпеки залежить від окремого користувача. Це не відбувається автоматично, але ви можете зробити це з вашої адміністративної панелі.

Крім того, існують також розширення для підвищення безпеки вашого вебсайту. Документація Joomla також містить контрольний список для запезпечення безпеки. Вона охоплює все, починаючи від налаштування і адмініструванн хостингу до відновлення з резервних копій. Щоб захистити свій сайт, це хороше місце для початку.

2.3 Drupal

Drupal — популярна вільна модульна система керування вмістом (СКВ) з відкритим вихідним кодом, написана на мові програмування PHP та розповсюджується за ліцензією GNU.[5]

Перша версія Drupal вийшла в 2001 році, і на сьогодні вона є третім найпопулярнішим рішенням для створення вебсайтів. Його загальна ринкова частка становить 2,2%, що означає, що близько 41 мільйона веб-сайтів використовують Drupal.

Цікаво, що ця система особливо популярна серед великих веб-сайтів. Серед одного мільйона найпопулярніших сайтів, він насправді більш популярний, ніж Joomla. Нижче ми поговоримо про причини цього.

Drupal також є найбільш технічно розвиненою CMS групи. Як і інші кандидати, його основною технологією є PHP, а також проект з відкритим кодом і програмним забезпеченням для спільноти.

Тепер, коли знаємо трохи більше про системи, про які ми говоримо, давайте порівняємо їх.

Drupal був створений для швидкої роботи. Отже, він використовує менше ресурсів, ніж його конкуренти, що є гарною новиною для витрат на сервер. Можливо, вам знадобляться деякі преміальні теми, які також коштують приблизно так само, як і для двох інших платформ.

З іншого боку, якщо сайт створює не розробник, створення сайту з Drupal значною мірою гарантує, що доведеться платити комусь. Це, безумовно, найбільш складне рішення в цьому списку і не підходить для початківців. З цієї причини витрати на розробку майже напевно повинні бути частиною вашого бюджету.

Звичайно, також можливо самостійно копатися в системі. Таким чином, користувач торгує часом за гроші. Залежно від того, наскільки цінним є час, це може бути гарною або поганою інвестицією. Навчання Drupal самому, швидше за все, значно подовжить запуск вашого сайту.

Установка Drupal працює так само, як і для згаданих раніше CMS. Крім того, CMS також пропонує так звані дистрибутиви, які є версією Drupal з попередньо встановленими розширеннями і модулями. Вони полегшують створення певних типів вебсайтів.

					IT41.100БАК.005 ПЗ	Лист
						23
Зм	Арк	№ докум.	Підпис	Дата		

Drupal Core Distributions Modules Themes

1,223 Distributions match your search

Maintenance status

Development status

Core compatibility

Status

Stability

Search Distributions

Sort by

Рисунок 2.13 – Дистрибутиви Drupal

Адміністративна область Drupal пропонує багато опцій налаштування з самого початку. Це дає вам великий контроль над своїм сайтом. На жаль, в минулому це також означало, що інтерфейс Drupal був найскладнішим. Проте, останнім часом були спроби його спростити. Адміністративну панель зображено на Рисунок 2.14.

Крім того, Drupal є найбільш технічно просунутим рішенням нашого порівняння CMS. Це також означає, що він вимагає знання PHP, HTML та інших мов програмування для здійснення будь-яких значущих змін. Це включає в себе оновлення сайту, який часто вимагає внесення коректив, щоб зробити існуючі компоненти сумісними з новою версією.

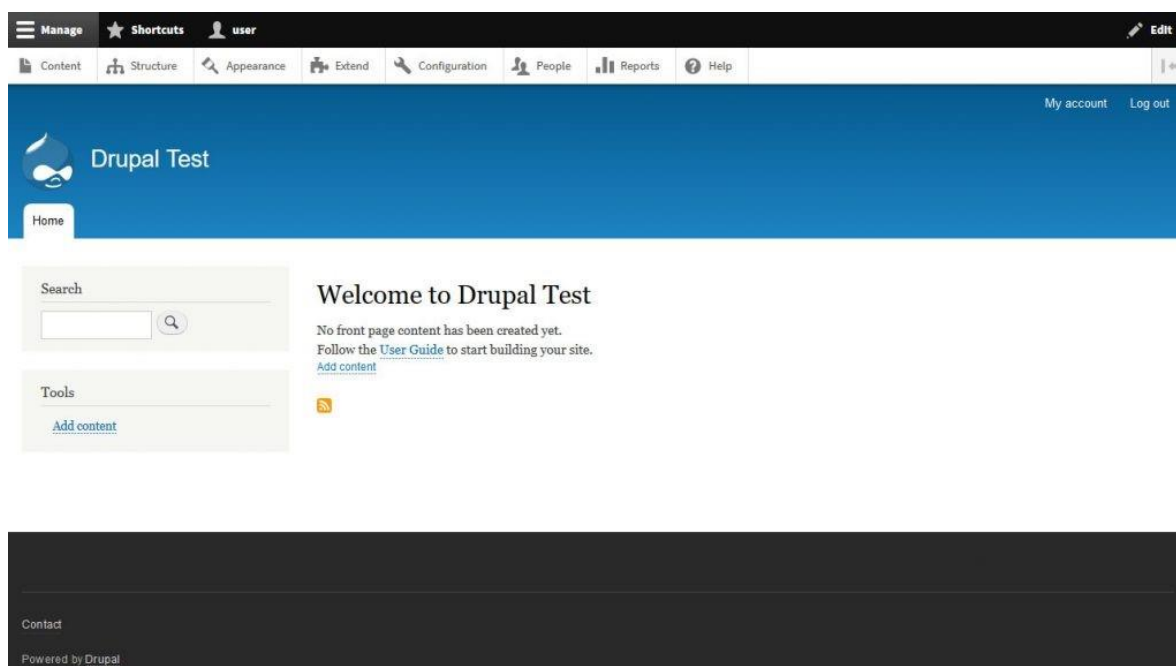


Рисунок 2.14 – Адміністративна панель Drupal

Отже, Drupal поставляється з найкрутішою кривою навчання і вимагає більшої кількості знань. Як ви можете собі уявити, це також ускладнює передачу сайту клієнтам.

Drupal також поставляється з циклом оновлення. Користувачі можуть розраховувати на нову версію кожні кілька місяців.

Якщо є технічні труднощі або питання, то тоді можливо покластися на підтримку спільноти у вигляді документації, форумів підтримки та груп користувачів. Спільнота активна і привітна, навіть якщо вона менша, ніж групи Joomla або WordPress.

Drupal - це все про створення власних веб-сайтів. З цієї причини, він поставляється з великою кількістю вбудованих опцій налаштування. Також можливо редагувати файли безпосередньо і налаштовувати майже все, що хочете. Отже, як розробник, є дуже мало, що ви не можете налаштувати.

Monthly

First Wednesday of every month	Bugfix release window for Drupal 8.4.x and 7.x
Third Wednesday of every month	Security release window for Drupal 8.4.x and 7.x

Current development cycle

Week of January 17, 2018	8.5.0-alpha1 released and 8.6.x-dev opened.
February 7, 2018	8.5.0-beta1 released. Final patch release window for 8.4.x (criticals only).
February 21, 2018	8.5.0-rc1 released. Final normal security release window for 8.4.x. (1)
March 7, 2018	8.5.0 released. End-of-life for Drupal 8.4.x. No other security releases. (1)
April 18, 2018	First 8.5.x-only security window (disclosure of 8.4.x security issues without backport). (1)

Next development cycle

Week of July 18, 2018	8.6.0-alpha1 released and 8.7.x-dev opened.
August 1, 2018	8.6.0-beta1 released. Final patch release window for 8.5.x (criticals only).
August 15, 2018	8.6.0-rc1 released. Final normal security release window for 8.5.x. (1)
September 5, 2018	8.6.0 released. End-of-life for Drupal 8.5.x. No other security releases. (1)
October 17, 2018	First security window for 8.6.x-only security window (disclosure of 8.5.x issues without backport). (1)

Рисунок 2.15 – Цикл оновлень Drupal

Крім того, як і інші CMS, Drupal також є частиною велочезної екосистеми. Він пропонує 40,000+ модулів і більше 2600 тем для додавання функціональності та варіантів дизайну на ваш сайт.

Завдяки застосованим стандартам кодування, вони також гарантують спільну роботу. Це не завжди так для плагінів WordPress і розширень Joomla. Їх установка набагато більш технічна, ніж з двома іншими CMS.

Безпека - це один з найсильніших сторін Drupal, а ця CMS дуже безпечна. Якщо виявиться уразливість, про це ви дізнаєтеся на офіційному сайті. Патчі будуть слідувати швидко.

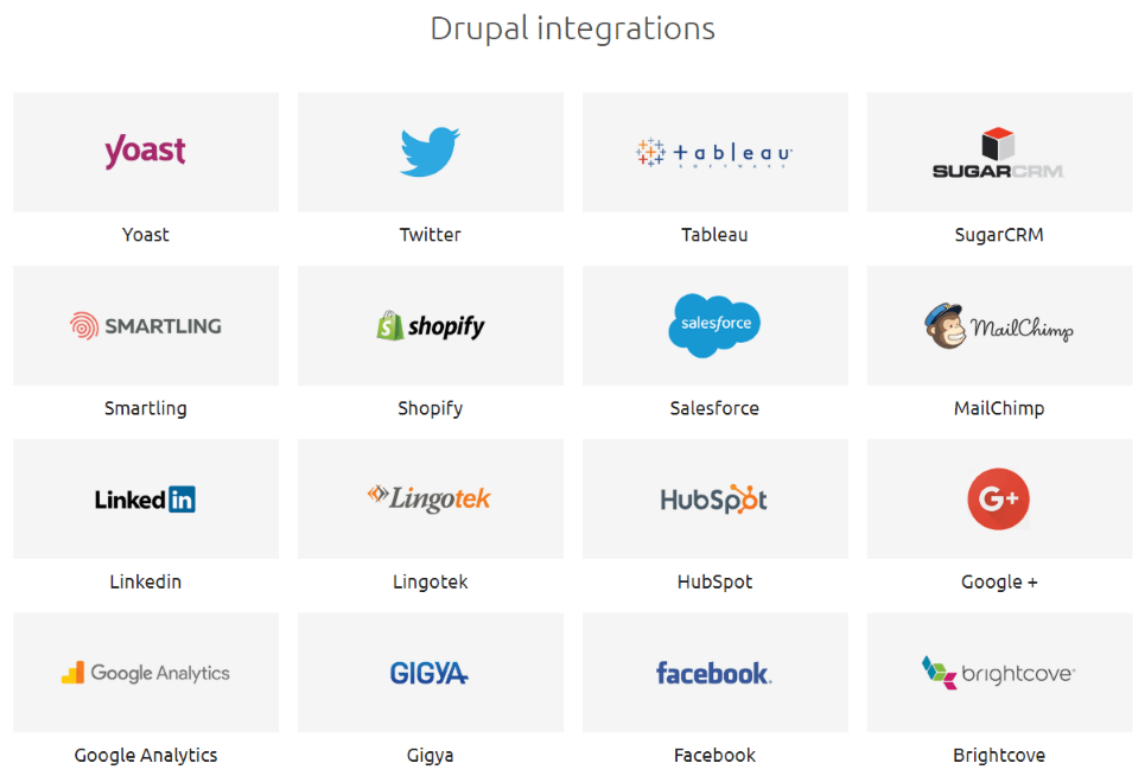


Рисунок 2.16 – Модулі, що інтегруються з Drupal

Крім того, Drupal пропонує розширення, щоб зробити ваш сайт більш безпечним. Наприклад, існує модуль, який створює звіти безпеки всередині адміністративної частини. Це допоможе виявити і усунути будь-які слабкі місця на сайті. Популярні плагіни наведено на Рисунку 2.16.

Drupal також відчув свою частку неприємностей, особливо в 2014 році, коли уразливість ін'єкції SQL призвела до того, що декілька веб-сайтів були зламані. Тим не менш, Drupal, як правило, відповідає його репутації.

Отже, всі вищерозглянуті CMS є дуже міцними засобами для розробки вебсайті. Однак більш доцільно буде використовувати фреймворки. Пов'язанно це з тим, що кожен з вищеперечислених варіантів має занадто великий надлишковий функціонал. В свою черге, це може привести до повільного завантаження сторінок, порівнянно з аналогічним рішенням на фреймворках і не дивлячись на те, що кожне з цих рішень оптимізовано.

					IT41.100БАК.005 ПЗ	Лист
						27
Зм	Арк	№ докум.	Підпис	Дата		

3 АНАЛІЗ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ

3.1 PHP

PHP (рекурсивний акронім словосполучення PHP: Hypertext Preprocessor) - це поширена мова програмування загального призначення з відкритим вихідним кодом. PHP спеціально сконструйований для веб-розробок та його код може впроваджуватися безпосередньо в HTML.[6]

PHP - це мова сценаріїв загального призначення, що використовується в основному для веброзробки для розширення веб-сторінок. Створений у 1994 році Расмусом Лерфордом, PHP почав працювати як набір скриптів, відомих як "Personal Home Page Tools". У 1995 році Расмус розширив свій пакет, щоб додати функціональність і випустив вихідний код для публіки.

На відміну від JavaScript, який є на стороні клієнта, PHP є серверною. Коли браузер запитує інформацію з сервера, сервер виконує код і відправляє результат клієнту. Як правило, вихідні дані є HTML-файлом, але це також може включати CSS і JavaScript. Браузер використовує цю інформацію для створення веб-сторінки.

PHP - це велика кількість корисних завдань, багато з яких ви відчули самі, коли ви відвідуєте популярні веб-сайти. Ці завдання допомагають розробникам додавати динамічний вміст і покращувати досвід користувачів.

PHP може використовуватися для взаємодії з файлами на сервері (створення, відкриття, читання, запис).

Також PHP може використовуватися для надсилання та отримання файлів cookie.

Ще PHP може використовуватися для доступу та зміни даних, що зберігаються в базі даних.

У додаток PHP може використовуватися для створення динамічного вмісту.

					IT41.100БАК.005 ПЗ	Лист
						28
Зм	Арк	№ докум.	Підпис	Дата		

PHP може використовуватися для створення областей, призначених тільки для користувачів, на вашому вебсайті та керування автентифікацією.

За допомогою PHP можуть бути створенні документів Excel і PDF.

Ви, ймовірно, відвідали сотні веб-сайтів, які використовують PHP. W3techs повідомляють, що 82,4% всіх веб-сайтів, на яких відома серверна мова програмування, використовують PHP, а ASP.Net - на далекому другому місці з 14,5%. Багато популярних CMS (системи управління контентом), включаючи Wordpress, Joomla і Drupal, широко використовують PHP.

PHP вважається більш легким для вивчення, ніж багато інших мов сценаріїв. Він використовує синтаксис C, що означає, що якщо ви коли-небудь вивчали C або C ++, то вам буде ще простіше підібрати.

Крім того, його популярність означає, що в Інтернеті доступний широкий спектр ресурсів. Наприклад, PHP є четвертою найпомітнішою темою на StackOverflow, з більш ніж 1,1 мільйонами тем і тисячами додаються щотижня.

PHP швидко розбирається і виконується, тому зниження продуктивності в порівнянні з завантаженням статичних файлів невелике. Оскільки PHP вбудований в HTML, він також швидко розробляється; це полегшує додавання PHP на існуючі веб-сайти без необхідності переписувати все.

PHP працює на багатьох платформах, сумісний з сучасними браузерами і підтримує широкий спектр баз даних.

Опитування розробників StackOverflow, проведене тисячами розробників, показує, що розробники використовують PHP у широкому діапазоні ситуацій.

PHP є п'ятою за популярністю технологією в цілому (25,9% респондентів використовують її).

PHP є третьою популярною мовою з розробниками інтерфейсу.

PHP є 4-ою за популярністю мовою з розробниками back-end.

PHP є четвертим за популярністю мовою з розробниками повного стека.

В основному, PHP дуже корисний у веб-розробці, незалежно від вашої загальної ролі.

					IT41.100БАК.005 ПЗ	Лист 29
Зм	Арк	№ докум.	Підпис	Дата		

Googoo повідомляє, що за місяць по всьому світу рекламується приблизно 8000 робочих місць, які вимагають РНР, із середньою зарплатою 66 000 доларів США. Тільки в США щомісяця рекламується приблизно 2500 робочих місць з середньою зарплатою в розмірі 89 000 доларів.

3.2 PostgreSQL

PostgreSQL вважається найсучаснішим і потужнішим SQL-сумісним і відкритим кодом-RDBMS. Він став першим вибором для корпорацій, які виконують складні та великі обсяги операцій з даними через його потужну технологію. Він використовує багатоваріантне управління паралелізмом (MVCC), яке дозволяє декільком авторам і читачам працювати в системі відразу. PostgreSQL надзвичайно здатна одночасно та ефективно обробляти декілька завдань, завдяки яким довіряють такі гіганти бізнесу, як Apple, Yahoo !, Instagram, Facebook та Instagram, а також численні інші телекомунікаційні, фінансові та державні установи.

PostgreSQL розроблявся з метою забезпечення відповідності стандартам, високої програмованості та розширюваності. Незважаючи на переконливу природу цієї СУБД, багато сторонніх інструментів і бібліотек роблять роботу з PostgreSQL простим. Інші особливості включають високу паралелізм, відповідність ACID (Атомічність, Консистенція, Ізоляція, Довговічність), об'єктне відношення, головним чином відповідність SQL і підтримку з боку активного співтовариства розробників.

СУБД MySQL з підтримкою Oracle є найвідомішим великомасштабним сервером баз даних. Це відкрита система, яка керує великою кількістю додатків і вебсайтів в Інтернеті. Легко налаштувати і вимагає мінімальної тонкої настройки для досягнення відмінних рівнів продуктивності. Інструменти стороннього графічного інтерфейсу, такі як Adminer, HeidiSQL, MySQL Workbench і dbForge Studio роблять MySQL ще більш простим для початку роботи з базою даних.

					IT41.100БАК.005 ПЗ	Лист
						30
Зм	Арк	№ докум.	Підпис	Дата		

Більш того, великий пул MySQL підручників і ресурсів, який є благом для початківців, тільки починаючи з бази даних.

MySQL є багатофункціональною системою і підтримує більшість функціональних можливостей SQL прямо або побічно, що очікується від СУБД. Вона дуже гнучка, що робить його популярним вибором для вебдодатків. Деякі інші функції включають доступність певних функцій безпеки, незалежність платформи, відповідність ACID та легкий доступ до підтримки.

PostgreSQL і MySQL є двома найбільш часто використовуваними СУБД, і рішення про вибір правильної системи управління реляційними базами даних для проекту може прискорити або сповільнити розробку. Незважаючи на те, що вони мають велику кількість спільних функцій, все це зводиться до вимог і того, як ви хочете підійти до проблеми, що стоїть перед нами. Нижче ми розглянемо найбільш видатні особливості обох систем і допоможемо вибрати найкращу СУБД.

Модель управління технологіями має унікальні переваги, а у випадку з MySQL та PostgreSQL демонструє значну різницю. PostgreSQL розроблена глобальною групою розвитку PostgreSQL і доступна під ліцензією з відкритим кодом, подібною до ліцензії MIT. Проект MySQL має свій вихідний код, відкритий для публіки під ліцензією GNU, разом з кількома угодами про власність. Зараз вона належить корпорації Oracle і повинна пропонувати різні платні версії для комерційного використання.

Стандарт відповідності SQL - це набір правил, яким повинна дотримуватися база даних при реалізації всіх директив і стандартів SQL. Відповідність SQL є значним обмеженням для компаній, які бажають працювати з гетерогенними базами даних для свого проекту (програми). PostgreSQL є найбільш сумісним з SQL, оскільки відповідає 160 з 179 основних функцій стандарту SQL, а також ряд додаткових функцій. З іншого боку, MySQL частково сумісний з SQL, оскільки не реалізує повний стандарт SQL. Тим не менш, він надає багато корисних функцій, які не належать SQL, таких як "відсутність обмежень CHECK".

Системи PostgreSQL і MySQL можуть працювати на Solaris, операційних системах Windows, Linux і OS X. OS X розроблена компанією Apple Inc. . PostgreSQL також підтримує ОС HP-UX, розроблену технологічним гігантом Hewlett-Packard, разом з відкритою ОС Unix. На відміну від цього, MySQL розширює підтримку відкритої ОС FreeBSD.

Якщо система підтримує різноманітні мови програмування, вона допомагає розробникам з різних фонів виконувати завдання мовою, якою вони зручні, таким чином підвищуючи ефективність. Оскільки сервер підтримує багато мов програмування для функцій бази даних, розробники можуть легко вирішити, чи виконувати завдання в клієнті або на сервері. Підтримка додаткових мов програмування надає розробникам додаткову потужність.

PostgreSQL був написаний на мові C і підтримує ряд мов програмування, найбільш подібних до C / C ++, Delphi, JavaScript, Java, Python, R, Tcl, Go, Lisp, Erlang і MySQL написаний на мовах C і C ++ і розширює підтримку C / C ++, Erlang, PHP, Lisp, Go, Perl, Java, Delphi, R і Node.js.

Безпека СУБД глибоко впливає на її надійність і виступає як одна з найкращих характеристик для пошуку оптимального вибору. Безпека баз даних вказує на колективне використання процесів і інструментів для захисту СУБД або бази даних від загроз і атак з незаконних джерел. MySQL є дуже безпечним і включає в себе безліч функцій безпеки, деякі досить продвинуті та встроєні в ядро. Він виконує протоколи безпеки на основі списків керування доступом (ACL) для операцій користувачів, таких як з'єднання та запити. PostgreSQL пропонує власну підтримку SSL для підключень для шифрування комунікацій клієнт / сервер. Більш того, вбудований розширення називається SE-PostgreSQL, який надає додаткові засоби контролю доступу на основі політики SELinux.

Обидва MySQL і PostgreSQL підтримують всі стандарти і мають кілька загальних методів доступу, які включають JDBC, ODBC і ADO.NET. JDBC - це база даних для доступу до API для програмування Java, тоді як ODBC - це стандартний API, який використовується для доступу до баз даних. ADO.NET - це

					IT41.100БАК.005 ПЗ	Лист
						32
Зм	Арк	№ докум.	Підпис	Дата		

набір API, який розробники використовували для отримання доступу до даних на основі XML. Для доступу до PostgreSQL можна також використовувати власну бібліотеку C і потокові API для платформи.

Реплікація баз даних відноситься до процесу електронного копіювання даних з одного комп'ютера або сервера на інший, що дозволяє всім користувачам мати доступ до однієї і тієї ж інформації. База даних може використовувати численні методи для зберігання надлишкових даних по декількох вузлах. MySQL використовує реплікацію master-master, в якій кожен вузол є майстром і має доступ до оновлення даних. Однак і PostgreSQL, і MySQL можуть виконувати реплікацію master-slave, в якій один вузол є майстром і керує зберіганням даних іншими вузлами. Інші типи реалізації можуть бути реалізовані в PostgreSQL за допомогою сторонніх розширень.

PostgreSQL підходить для використання у великих системах, де необхідно аутентифікувати дані, а швидкість читання / запису є критично важливою для успіху. Крім того, він також підтримує ряд підсилювачів продуктивності, які доступні тільки в пропрієтарних рішеннях, включаючи паралельність без блокування читання, SQL-сервер і підтримку геопросторових даних серед багатьох інших. Загалом, PostgreSQL найкраще підходить для систем, які вимагають виконання складних запитів, або сховищ даних і аналізу даних.

MySQL є першим вибором для тих веб-проектів, які вимагають бази даних лише для транзакцій даних, а не щось складне. Вона працює виключно добре в системах онлайн-аналітичної обробки (OLAP) та онлайн-обробки транзакцій (OLTP), де потрібні тільки високі швидкості читання. Тим не менш, MySQL почне неефективно, як тільки це буде підкреслено важкими навантаженнями або складними запитами.

Хоча ми обговорюємо, де ці системи можуть бути використані, фактична продуктивність СУБД може бути виміряна тільки шляхом оцінки метрики можливих сценаріїв, оскільки вона головним чином залежить від вимог і характеру проекту, для якого він використовується.

Суспільна підтримка є важливою для поліпшення будь-якої системи бази даних. PostgreSQL має дуже активну спільноту, яка постійно допомагає в покращенні існуючих функцій, її винахідливі комітатори докладають всіх зусиль, щоб переконатися, що база даних залишається найбільш передовою базою даних з останніми функціями і максимальною безпекою. У MySQL також є велика спільнота послідовників і учасників, які зосереджують увагу на збереженні існуючих функцій, а іноді з'являються нові функції.

3.3 Фреймворк

PHP є однією з найпопулярніших мов програмування в усьому світі, і нещодавній випуск PHP 7 зробив цей серверний мова програмування кращим і більш стабільним, ніж будь-коли.

PHP широко використовується у великих проектах. Facebook, наприклад, використовує PHP для підтримки та створення своїх внутрішніх систем. WordPress використовує PHP для створення внутрішніх проектів, які в свою чергу включають в себе більше 26% інтернету. В даний час, PHP наділяє понад 82% веб-сайтів.

Ми розглянемо три найпопулярніших фреймворка PHP: Symfony, Laravel і Yii. Ми розглянемо, як вони порівнюються, щоб вибрати коректне рішення для реалізації.

Який сенс використовувати фреймворк замість сирого PHP для розробки вашої програми? Кілька переваг використання рамки включають:

- фреймворки PHP роблять розвиток швидше. Наприклад, вам не потрібно писати складні запити для отримання даних з бази даних;
- фреймворки PHP забезпечують операції CRUD (створення, читання, оновлення та видалення);
- фреймворки дозволяють розробникам легко масштабувати системи;

- підтримка коду простіше, ніж при застосуванні нативного PHP. Код програми є коротким і легким для роботи;
- модель MVC забезпечує швидкий розвиток;
- фреймворки допомагають краще забезпечити вебдодаток від поширених загроз безпеки;
- принцип не повторювати себе (DRY) гарантує, що мінімальний код має максимальний вплив.

Перераховані вище переваги занадто значні, щоб їх можна було ігнорувати. Незважаючи на те, що сировину PHP можна використовувати для створення будь-якої програми, поточні стандарти розвитку вимагають інструментів і навичок керування часом для задоволення ринкового попиту.

Для вибору найкращого фреймворку для певної задачі необхідно розглянути наступні пункти, а саме:

- які особливості та функціональність фреймворку;
- яка крива навчання фреймворку;
- наскільки масштабованим є фреймворк;
- чи активно розвивається та підтримується базовою командою;
- чи забезпечує система довгострокову підтримку (підтримка LTS);
- чи підтримує цей механізм сильну підтримку громади;

Symfony - це набір багаторазових компонентів PHP, що дозволяє розробникам створювати масштабовані, високопродуктивні програми[7]. З 30 компонентів, з яких вибирати, розробник має повну свободу експериментувати і працювати в середовищі RAD. API Symfony також дозволяють легко інтегруватися з додатками інших виробників, і його можна використовувати з популярними фронтальними фреймворками, такими як AngularJS.

Багато популярних проектів, включаючи Drupal і phpBB, також використовують фреймворк Symfony. Фактично, Laravel, найпопулярніший фреймворк PHP, побудований на основі Symfony.

Laravel, відомий як "PHP фреймворк для вебремісників"[8], пропонує відмінну спільноту і перемагає як найбільш популярний фреймворк. Одним з провідних розробників Laravel на Livecoding.tv є Sfikell.

У травні 2015 року Ларавель оголосив, що версія 5.1 запропонує довгострокову підтримку протягом двох років. Версія 5.2 була розгорнута в грудні 2015 року. Багато хостингові компанії надають підтримку Laravel і пропонують хостинг рішення для додатків Laravel.

Створений Цян Сюе в 2008 році, Yii - це безпечна, швидка, високопродуктивна програма для розробки додатків / веб-розробки[9]. Yii використовує менеджер залежностей Composer для PHP для обробки різних залежностей і установок (про це пізніше). Yii також є найшвидшою фреймворком PHP, завдяки застосуванню lazy.

Ще одна цікава особливість Yii - інтеграція jQuery. Інтеграція дає змогу розробникам front-end швидко прийняти рамки, і вона використовує риштування для створення коду. Подібно до Symfony, Yii також використовує компоненти для швидкої розробки додатків.

Всі три фреймворки чудово підходять для побудови додатків Web 2.0, але кожна з них має іншу мету. Давайте подивимося на їхні особливості та те, як вони складаються.

Двигуни шаблонів мінімізують зусилля розробників і забезпечують кращу функціональність при написанні інтерфейсного коду. Двигуни шаблонів забезпечують такі функції, як автоматичне витікання HTML-файлів і фільтри, а також заповнюють прогалину, що залишилася від сирого PHP.

Twig - це сучасна система шаблонів для PHP. Symfony використовує Twig в свою користь і дозволяє розробникам писати чистий, лаконічний код і можливість робити більше, ніж з необробленим PHP. Twig робить це з наступним кодом:

```
{{ var }} {{ var | escape }} {{ var | e }} {# ярлик, щоб уникнути змінної #}
```

Можна ознайомитися із сайтом Twig, щоб дізнатися більше про його функції.

На відміну від інших систем шаблонів, Blade дозволяє використовувати PHP-код у поданнях. Крім того, Blade має нульову навантаження на продуктивність програми, оскільки файли вигляду зберігаються у розширенні `.blade.php`. Весь код у файлах перегляду перетворюється на необроблений PHP під час обробки.

За замовчуванням Yii не використовує будь-яку систему шаблонів третіх сторін, але це не означає, що їй бракує підтримки системи шаблонування. Вибір системи шаблонів залежить від команди розробників. Рекомендується використовувати Twig і Smarty. Symfony використовує Twig, тому, якщо ви використовували Symfony в минулому, ви можете використовувати Twig для вашого наступного проекту Yii.

Тут немає чіткого переможця. Всі три фреймворки використовують двигуни-шаблони для кращого кодування та обслуговування. Невеликою перевагою для Yii є те, що фреймворк не мають заздалегідь визначеної системи шаблонів.

Кожна структура створена по-різному. Symfony працює на багаторазових компонентах і забезпечує кращу модульність. Symfony також використовує модель і контролер для розробки веб-додатків, які можуть виглядати іржавими для багатьох нових розробників, але він працює. Крім того, Symfony є гарним прикладом модульної структури. Модульно можна використовувати 30 компонентів, наданих Symfony у вашому проекті.

Yii використовує структуру MVC. (Symfony надає підтримку MVC, яка більш детально обговорюється в розділі Symfony4 MVC на сайті blog.sznepka.pl.)

Symfony може використовуватися для швидкого розвитку і складних проектів. Незважаючи на те, що існує дискусія про те, які рамки краще для складних проектів, Symfony демонструє блискучу обробку складності порівняно з іншими системами. Yii також використовує компоненти, але не є такою модульною, як Symfony. Laravel не надає модульного підходу, настільки різкого, як два інших рамки.

Якщо шукаємо модульні фреймворки, зверніться до Symfony. Інакше, Laravel і Yii є великим вибором.

Три фреймворка забезпечують багаторазові процедури встановлення. Якщо ви використовуєте Composer для обробки пакетів, ви будете раді дізнатися, що всі фреймворки можуть бути встановлені через Composer.

Для Symfony роль композитора є більш важливою. Ідея обробки компонентів найкраще виконується за допомогою менеджера залежностей PHP Composer.

Існують інші способи встановлення відповідних фреймворків. Наприклад, можна встановити фреймворки, використовуючи простий метод архівування.

Після інсталяції Yii надає веб-додаток і основний шаблон для роботи. Symfony 4 також надає демо-додаток для початку роботи.

Laravel також легко встановлюється за допомогою Composer create-project або через Laravel Installer.

З точки зору компанії або клієнта, важливим є швидке отримання заявки на ринок для задоволення споживчого попиту і подолання конкурентів. Symfony виступає за надійну структуру з сильною спільнотою, що стоїть за нею. Laravel швидко зростає, але все ще є шлях, перш ніж вважатися де-факто вибором для розробки PHP. З іншого боку, якщо не знаємо жодних PHP-фреймворків і хочемо якомога швидше почати роботу, розгляньте Laravel. Laravel має легку криву навчання, і знайдете багато навчальних посібників онлайн, щоб допомогти розпочати роботу. Yii виконує роботу на наступний рівень і надає кодові приклади для більш швидкого створення та розробки коду.

Виконання будь-якої програми має значення лише в тому випадку, якщо це програма реального часу з використанням критичних даних. Необхідно оцінити скільки вебдодатків залежить від високої продуктивності? Не багато, але використання фреймворків може відігравати вирішальну роль у багатьох проектах.

					IT41.100БАК.005 ПЗ	Лист
						38
Зм	Арк	№ докум.	Підпис	Дата		

Соціальні мережі є яскравим прикладом подій у реальному часі. Коли йдеться про вибір кращого фреймворку для кодування високопродуктивних додатків, Yii виділяється як найшвидший фреймворк PHP у данному огляді.

Продуктивність Laravel є дуже дискусійною. Це повільніше, але чи має значення це. Ви знайдете онлайнові ресурси для прискорення роботи, включаючи керівництво по GitHub для швидшого застосування програми Laravel.

Symfony 4 пропонує кращу підтримку бази даних. Можна працювати з масивом баз даних, включаючи NoSQL і DynamoDB. Yii і Laravel також корисні в цьому відношенні, але вони підтримують менше баз даних, ніж Symfony.

3.4 Composer

Composer є відносно новим і вже досить популярним менеджером залежностей для PHP. Ви можете описати, з яких бібліотек залежить ваш проект, і Composer встановлює для вас необхідні бібліотеки. Більш того, Composer не є менеджером пакетів в класичному сенсі[10]. Так, він працює з об'єктами, які ми будемо називати «пакунками» або бібліотеками, але вони встановлюються всередині кожного проекту окремо, а не глобально (це одна з головних відмінностей від старого доброго PEAR).

Коротко, як це працює:

- є проект, який залежить від декількох бібліотек;
- деякі з цих бібліотек залежать від інших бібліотек;
- У проекті ви описуєте ті бібліотеки, від яких безпосередньо залежить ваш код.

Composer знаходить необхідні версії необхідних бібліотек для всього проекту, завантажує їх і встановлює в папку проекту.

Створюючи Composer, автори отримали ідеї та натхнення від подібних проектів: npm для Node.js та Bundler для Ruby.

Спочатку він був розроблений і розроблявся двома людьми Нілсом Адерманом і Джорді Боджано, тепер у проекті беруть участь більше двадцяти учасників. Проект написаний на PHP 5.3, поширюється під ліцензією MIT і доступний на github.

Перші зобов'язання були зроблені в квітні 2011 року, і сьогодні Composer знаходиться на етапі 1.7 версії. І вона вже досить стабільна і використовується багатьма популярними проектами PHP (наприклад, Symfony 4). Список проектів, що використовують Composer, можна переглянути на сайті packagist.org - це офіційний репозиторій композицій пакетів[15].

У цьому розділі описано основні можливості Composer і у практичній частині будемо використовувати саме його для керування залежностями у проекті.

За замовчуванням пакунки завантажуються з офіційного сховища packagist.org. Кожен може вільно додати свій пакет для того, щоб зробити його встановлення легким і зручним для всього світу;

Пакети можна завантажити не тільки з packagist.org, але і з будь-якого сховища git, mercurial або svn;

При завантаженні пакунків з github.com або bitbucket.org не потрібна встановлена система контролю версій (git або hg), композитор працює через API цих сайтів;

Репозиторій git / hg / svn з пакетом може бути розташований не тільки на одному з перелічених вище сайтів, але й у будь-якому іншому місці, наприклад, у локальній мережі підприємства або взагалі на локальному жорсткому диску;

Крім того, встановлена бібліотека не повинна бути у вигляді пакета Composer, яку можна встановити з будь-якого сховища git / hg / svn довільної структури;

Нарешті, встановлений пакет не повинен бути сховищем git / hg / svn, це може бути довільний zip-файл, доступний на будь-якому uri!

Всі пакети встановлені в поточному каталозі (звідки виконувалася команда установки), це дозволяє мати кілька різних версій бібліотек при паралельній роботі над різними проектами;

Команда оновлення оновлює всі встановлені (або повторно встановлює випадково видалені) пакети в нові версії. Або він може не оновлювати версії до найсвіжіших, якщо ви створили спеціальний файл `composer.lock` - це дозволяє виправити комбінацію стабільних версій усіх бібліотек, що використовуються в проекті;

Після встановлення пакунків автоматично генерується `autoload.php`, за допомогою якого ви можете підключити встановлені бібліотеки до коду проекту. Під час підготовки пакета Composer рекомендується використовувати PSR-0, стандарт для розміщення і назв файлів `php`, щоб автозавантаження могло легко їх знайти. У будь-якому випадку, автор пакета може описувати правила, за допомогою яких автозавантаження буде шукати файли різних класів або `namespaces`. Якщо ви встановите бібліотеку, яка не обрамлена як пакет композиції (наприклад, довільний сховище `git` з `github`), то завдання опису правил автозавантаження лягає на ваші плечі. Таким чином, не існує магії з генерованим `autoload.php` - він може завантажити все (навіть бібліотеки з набором функцій поза класами), поки правила описуються (бібліотекою або вами).

					IT41.100БАК.005 ПЗ	Лист
						41
Зм	Арк	№ докум.	Підпис	Дата		

4 ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЕКТУ

4.1 Проектування схеми бази даних

Проектування баз даних - це організація даних відповідно до моделі бази даних. Конструктор визначає, які дані повинні зберігатися і як взаємозв'язані елементи даних[11].

Проектування бази даних включає класифікацію даних та виявлення взаємозв'язків. Це теоретичне представлення даних називається онтологією. Онтологія є теорією дизайну бази даних.

У більшості випадків особа, яка займається розробкою бази даних, є людиною, яка має досвід роботи в області проектування баз даних, а не знання в області, з якої дані, що підлягають збереженню, накреслені, наприклад. фінансова інформація, біологічна інформація тощо. Таким чином, дані, які потрібно зберігати в базі даних, повинні визначатися у співпраці з особою, яка має досвід роботи в цій галузі, і хто знає, які дані повинні зберігатися в системі.

Цей процес є тим, який, як правило, вважається частиною аналізу вимог, і вимагає навичок з боку розробника бази даних, щоб отримати необхідну інформацію від тих, хто володіє знаннями домену. Це пояснюється тим, що ті, у кого необхідні знання в області часто не можуть чітко виразити, які їхні системні вимоги до бази даних є такими, якими вони не звикли мислити з точки зору дискретних елементів даних, які необхідно зберігати. Дані, що зберігаються, можуть бути визначені специфікацією вимог.

Після того, як проектувальник баз даних знає дані, які повинні зберігатися в базі даних, вони повинні визначити, де знаходиться залежність у даних. Іноді при зміні даних можна змінювати інші невидимі дані. Наприклад, у списку імен і адрес, припускаючи ситуацію, коли кілька людей можуть мати однакову адресу, але одна людина не може мати більше однієї адреси, адреса залежить від імені. При наданні імені та списку адреса може бути однозначно визначена; однак,

					IT41.100БАК.005 ПЗ	Лист
						42
Зм	Арк	№ докум.	Підпис	Дата		

зворотне не виконується - коли задається адреса і список, ім'я не може бути однозначно визначене, оскільки кілька людей можуть перебувати в адресі. Оскільки адреса визначається ім'ям, адреса вважається залежною від імені.

Як тільки зв'язки і залежності між різними елементами інформації були визначені, можна організувати дані в логічну структуру, яка потім може бути відображена в об'єкти зберігання, які підтримуються системою управління базами даних. У випадку реляційних баз даних об'єктами зберігання є таблиці, які зберігають дані в рядках і стовпцях. У об'єктній базі даних об'єкти зберігання відповідають безпосередньо об'єктам, які використовуються об'єктно-орієнтованою мовою програмування, що використовується для запису додатків, які будуть керувати та отримувати доступ до даних. Відносини можуть бути визначені як атрибути задіяних класів об'єктів або як методи, що працюють на класах об'єктів.

Спосіб виконання цього відображення такий, що кожен набір пов'язаних даних, які залежать від одного об'єкта, реального чи абстрактного, поміщається в таблицю. Відносини між цими залежними об'єктами потім зберігаються як зв'язки між різними об'єктами.

Кожна таблиця може представляти реалізацію або логічного об'єкта, або відносини, що об'єднує один або більше екземплярів одного або більше логічних об'єктів. Відносини між таблицями можуть зберігатися як посилання, що зв'язують дочірні таблиці з батьками. Оскільки складні логічні відносини самі є таблицями, вони, ймовірно, будуть мати посилання на більш ніж одного з пов'язаних таблиць.

Спираючись на теоретичні відомості по проектуванню баз даних, для постановленої задачі можна виділити наступні основні сутності:

- User – користувач
- Profile – дані користувача
- Job – місце роботи
- Education – місце навчання

Розглянувши ці сутності, нам необхідно встановити зв'язки між ними та визначити типи зв'язків:

- User – Profile (1:1)
- Profile – Job (1:n)
- Profile – Education (1:n)

Кожен користувач системи може мати лише одну сторінку профілю і не більше.

Кожен користувач може мати декілька місць роботи у блоці з навичками, але оскільки всю інформацію ми зберігаємо у таблиці з профілями, то місця роботи ми повинні пов'язувати з профілем. Те ж саме відноситься і до місць навчання.

На основі виділених сутностей необхідно виділити дані, що належать до цих сутностей. Так ми зможемо побудувати ER-діаграму, що наведено на Діграмі 4.

Далі нам необхідно які саме ми дані будемо зберігати у кожній таблиці. Почнемо з таблиці для користувачів.

Users:

id – унікальний ідентифікатор

email – електронна пошта для авторизації

password – пароль для авторизації

slug – значення, що буде використовуватися для адреси профілю

created_at – дата створення

updated_at – дата оновлення

deleted_at – дата видалення, за змовчуванням NULL

Profiles:

id – унікальний ідентифікатор

user_id – зовнішній ключ на користувача

email – електронна пошта для показу на сторінці

tel – номер телефону

avatar – посилання на зображення

first_name – поле для імені

last_name – поле для прізвища

profession – посада

general – головна інформація користувача

publication – поле для збереження інформації про публікації

Jobs:

id – унікальний ідентифікатор

profile_id – зовнішній ключ на профіль

title – назва закладу

position – посада

start – початок роботи

end – закінчення роботи

description – опис задач на посаді

Education:

id – унікальний ідентифікатор

profile_id – зовнішній ключ на профіль

title – назва закладу

field – напрям навчання

start – початок навчання

end – закінчення закладу

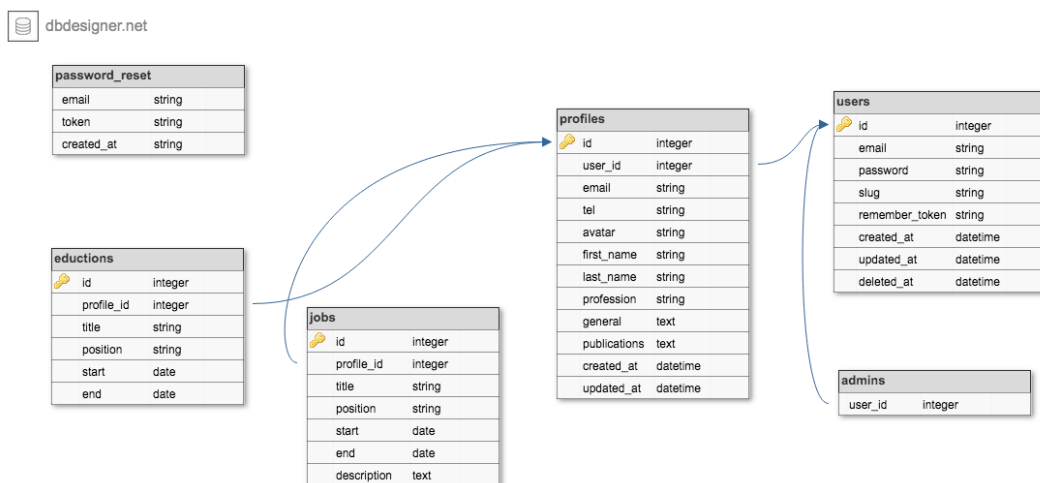


Рисунок 4.1 – Схема бази даних

Отже, було виділено основні сутності, дані та зв'язки між сутностями та даними.

4.2 Маршрутизація

Маршрутизація - процес визначення маршруту даних в мережах зв'язку.

Маршрутизація в мережі інтернет заснована на протоколах TCP / IP.

Передача інформації здійснюється за допомогою IP-пакетів, заголовок кожного IP-пакета містить IP-адреси одержувача і відправника пакету[12]. Кожен пакет обробляється маршрутизатором відповідно до його таблицею маршрутизації. Таблиця, в свою чергу, містить інформацію, комп'ютера з яким адресою направляти пакети з тим чи іншим діапазоном адрес. Наприклад, всі пакети певного діапазону можуть направлятися іншому маршрутизатора, який «відповідає» за цей сегмент.

У ряді випадків маршрутизатор може перетворювати заголовок пакета, замінюючи адреси відправника та / або одержувача пакета. Зокрема, це відбувається при взаємодії локальної мережі (що має свої адреси) з глобальною мережею Інтернет. В цьому випадку локальна мережа може бути видна ззовні по одному глобальному IP-адресою[13]. Для того, щоб маршрутизатор міг направляти пакети з одним глобальним адресою тим чи іншим одержувачам в локальній мережі, використовується таблиця NAT, де крім IP-адрес вказуються порти, що ідентифікують додатки, що встановлюють з'єднання. При цьому номери портів вказані не в заголовку IP-пакета, а в заголовку сегмента TCP або UDP (сегменти інкапсулюються в поле даних IP-пакетів). Це дозволяє здійснювати взаємно-однозначну ідентифікацію одержувача і відправника в тих випадках, коли за одним глобальним адресою знаходиться безліч комп'ютерів локальних мереж.

Систему можна поділити на дві частини: загальну та адміністративну. До адміністративної відноситься частина, у якій головні користувачі, а саме

					IT41.100БАК.005 ПЗ	Лист
						46
Зм	Арк	№ докум.	Підпис	Дата		

адміністратори, можуть керувати усіма користувачами та їх даними. Але загальну частину ми також можемо поділити на дві частини. У одній ми даємо доступ всім відвідувачам нашого рішення отримувати доступ та інформацію про користувачів з профілю, а у іншій частині авторизовані користувачі можуть керувати своїми тільки своїм профілем.

Отже, ми можемо виділити такі основні шляхи:

/login -- шлях для авторизації

/slug -- шлях для перегляду профілю

/slug/admin – шлях для редагування профілю

/admin/users – шлях для керування усіма користувачами

/admin/profiles – шлях для керування усіма профілями

Необхідно також зазначити, що за стандартами протоколу HTTP, можливо використовувати для одних і тих самих шляхів різні методи, а саме:

GET – для одержання даних

POST – для створення даних

PUT – для зміни даних

DELETE – для видалення даних

Отже, дійсна кількість шляхів може відрізнятися від наведених вище.

4.3 Контролери

При розробці даного програмного забезпечення були вирішено розміщувати усю логіку обробки запитів саме у контролерах. Пов'язано це з тим, що винесення коду до сервісів може зменшити швидкість розробки та не надасть необхідної гнучкості, хоча такий підхід і є більш правильним.

Основна логіка для відображення профілей розміщена у ProfileController. Він включає у себе метод для відображення, збереження, зміни. Давайте більш детально розглянемо метод для відображення.

Як аргумент метод для відображення приймає у себе slug, що отримується із адреси. Але slug міститься у таблиці з користувачами, у той час як нам необхідно отримати сам профіль. З минулих розділів ми знаємо, що між користувачами та профілями зв'язок 1 до 1. Отже, для отримання профілю нам необхідно використати наступний код:

```
$profile = User::where("slug", $slug)->first()->profile()->first();  
if(!$profile)  
    abort(404);
```

Якщо профілю не існує, то будемо повертати помилку 404.

Для того, щоб завантажити місця роботи та освіти, нам необхідно завантажити зв'язки. Також для кожного отриманого запису необхідно замінити повноцінну дату лише на рік.

```
$jobs = $profile->jobs()->get();  
foreach ($jobs as $job) {  
    $job->start = Carbon::parse($job->start)->year;  
    $job->end = $job->end !== null ? Carbon::parse($job->end)->year :  
    "Present";  
}
```

Кожен контролер повинен повертати об'єкт класу Response, але оскільки ми у відповіді повинні повернути шаблон, то ми можемо використати функцію та прив'язати до відповіді необхідні змінні.

```
return view('profile.index')->with([  
    'profile' => $profile,  
    'jobs' => $jobs,  
    'educations' => $educations,  
]);
```

Логіка для створення та зміни дуже схожа, за винятком того, що при редагуванні нам необхідно використовувати slug. Тому розглянемо лише один із цих методів.

Всю необхідну інформацію для профіля ми збережемо у одному масиві:

```
$profileData = [  
    'user_id' => Auth::id(),  
    'first_name' => $request->first_name,  
    'last_name' => $request->last_name,  
    'profession' => $request->profession,  
    'email' => $request->email,  
    'tel' => $request->phone,  
    'avatar' => $request->filepath,  
    'general' => $request->information,  
    'publications' => $request->publications,  
];
```

Наступним кроком нам необхідно перевірити наявність профілю і якщо його не існує – ми створимо новий, інакше збережемо у попередньостворений.

```
$profile = Profile::where('user_id', Auth::id())->first();  
  
if($profile)  
    $profile->update($profileData);  
else  
    $profile = new Profile($profileData);  
$profile->save();
```

Далі необхідно зберегти місце роботи та освіти. Але у тому випадку, коли не існує створеного місця роботи або освіти, то створюється нові. Для того, щоб відмітити, що вже існує створена сутність, ми перевіряємо наявність ідентифікатора. Саме це і є критерієм створення.

```
foreach ($request->educations as $item){  
    $educationData = [  
        'profile_id' => $profile->id,  
        'title' => $item['title'],  
        'field' => $item['field'],
```

```

        'start' => Carbon::create($item['start']),
        'end' => $item['end'] === null || $item['end'] === "" ? null :
Carbon::create($item['end']),
    ];
    if(isset($item["id"]))
        $education = Education::find($item["id"])->update($educationData);
    else
        $education = (new Education($educationData))->save();
    }

```

Далі розглянемо контролер для адміністратора. Для адміністратора було створено два контролери: для керування користувачами та профілями.

Для перевірки того, що користувач є адміністратором створимо міدلвайр, що перевіряє чи є користувач адміністратором. Для створення міدلвайра ми використовуємо команду у консолі.

```
php artisan make:middleware Admin
```

Сама перевірка є дуже простою та використовує метод з моделі для перевірки адміністратора. Якщо користувач не є адміністратором, то користувач буде перенаправлений на головну сторінку.

```

if(!$request->user()->isAdmin()) {
    return redirect('/');
}
return $next($request);

```

Методи для створення та зміни користувачів та профілями для адміністраторів є дуже простими. Наприклад метод для збереження користувачів буде виглядати наступним чином:

```

public function store(UserCreateRequest $request)
{
    $user = new User([
        "email" => $request->email,

```

```

        "password" => bcrypt($request->password),
        "slug" => $request->slug,
    ];

    $user->save();

    return redirect()->route("admin.users.index");
}

```

Отже, було побудовано основні методи для контролерів для керування різноманітними сутностями.

4.4 Представлення

Представлення – це шар програмного забезпечення, що відповідає за зовнішній вигляд продукту.

Blade - це простий, але потужний двигун шаблонів, забезпечений Laravel. На відміну від інших популярних PHP-шаблонів, Blade не обмежує використання звичайного PHP-коду у вашому перегляді. Насправді, всі погляди Blade компілюються в звичайний PHP-код і кешуються, поки вони не будуть змінені, тобто Blade додає по суті нульові накладні витрати на вашу програму. Файли вигляду леза використовують розширення `.blade.php` і зазвичай зберігаються в каталозі `resources / views`.

У данному проекті було реалізовано макет, який будемо доповняти за допомогою контенту та скриптів. Для доповнення, ми розмітили три точки, які нам доступні.

```

<main id="content" class="container-fluid">

    @yield('content')

</main>

@stack('css')

```

@stack('js')

Найскладнішою є сторінка показу профіля, тому на її прикладі і будемо розглядати.

Для виводу перемінних у шаблон використовується спеціальна конструкція. Далі наведено приклад виводу верхнього блоку з ім'ям користувача.

```
<div class="name">
    {{ $profile->last_name }}
    <span>{{ $profile->first_name }}</span>
</div>
<div class="degree">
    {{ $profile->profession }}
</div>
```

Далі вся ліва частина дуже схожа, тому її розглядати не будемо. Права частина виглядає наступним чином.

```
<div class="col-md-8 col-xs-12 content">
    @include('profile.partial.experience', ['jobs' => $jobs])
    @include('profile.partial.education', ['educations' => $educations])
    @include('profile.partial.general', ['general' => $profile->general])
    @include('profile.partial.publications', ['publications' => $profile-
>publications])
</div>
```

Як бачимо, для зменшення розміру і зручності було поділено один великий файл на частини, а потім включаємо ці файли у результуючий шаблон. Під час підстановки ми передаємо змінні, що будуть використовуватися.

Отже, у данній главі побудовано програмне забезпечення та розглянули основні моменти, з якими зіштовхнулися під час розробки.

5 ПРОЦЕС ДЕПЛОЮ ТА ПРИКЛАД ВИКОРИСТАННЯ

5.1 Деплой

Для деплою розробленого програмного забезпечення необхідно операційна система Debian або Ubuntu. Далі нам необхідно, щоб були встановлені наступні пакети:

- Git;
- PHP 7.3;
- Nginx;
- PostgreSQL;
- Composer;
- Node 12+.

Необхідно налаштувати сервер nginx та створити папку для розміщення проекту. Далі клонуємо репозиторій з проектом.

```
git clone https://github.com/Yunicot/acts-blog.git
```

Переходимо у новостворену папку з проектом.

```
cd acts-blog/
```

У нашому проекті ще немає завантажених пакетів, які потребуються для роботи. Для їх завантаження необхідно виконати наступну команду:

```
composer i
```

```
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: barryvdh/laravel-debugbar
Discovered Package: barryvdh/laravel-ide-helper
Discovered Package: beyondcode/laravel-dump-server
Discovered Package: fideloper/proxy
Discovered Package: intervention/image
Discovered Package: jeroennoten/laravel-adminlte
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Discovered Package: unisharp/laravel-filemanager
Package manifest generated successfully.
```

Рисунок 5.1 – Інсталювання пакетів

					IT41.100БАК.005 ПЗ	Лист
						53
Зм	Арк	№ докум.	Підпис	Дата		

Приклад вдалого виконання команди наведено на Рисунку 5.1.

Зараз програмне забезпечення вже повинно відкриватися, але з помилкою. Далі нам необхідно провести міграції. Для цього ми виконуємо команду:

```
php artisan migrate --seed
```

Це також створить нам нашого головного користувача та дає йому права адміністрування.

```
Dropped all tables successfully.  
Migration table created successfully.  
Migrating: 2014_10_12_000000_create_users_table  
Migrated: 2014_10_12_000000_create_users_table  
Migrating: 2014_10_12_100000_create_password_resets_table  
Migrated: 2014_10_12_100000_create_password_resets_table  
Migrating: 2019_06_04_121101_create_profiles_table  
Migrated: 2019_06_04_121101_create_profiles_table  
Migrating: 2019_06_04_172421_create_jobs_table  
Migrated: 2019_06_04_172421_create_jobs_table  
Migrating: 2019_06_04_193355_create_educations_table  
Migrated: 2019_06_04_193355_create_educations_table  
Migrating: 2019_06_04_194345_create_admins  
Migrated: 2019_06_04_194345_create_admins  
Seeding: UserSeeder  
Database seeding completed successfully.
```

Рисунок 5.2 – Виконання міграцій

На данному етапі в нас повинна відкриватися головна сторінка без помилок, але ще залишилося зібрати зовнішній вигляд. Нам для цього також потрібно встановити пакети, а у додаток скопіювати файли стилів та скриптів.

```
npm i
```

Приклад виконання команди наведено на Рисунку 5.3.

```
npm run prod
```

Якщо команда буде виконано успішно, то результат буде такий самий, як на Рисунку 5.4.

Після цього буде створена нова папка `node_modules` та саме вона буде містити у собі всі необхідні бібліотеки для фронт-енду.

Нижче наведено приклад, як саме повинно виглядати вікно для інсталювання та збору, а також результат, який ми повинні отримати.

```
node-pre-gyp WARN Using needle for node-pre-gyp https download
[fsevents] Success: "/Users/vantino/Projects/acts-blog/node_modules/fsevents/lib/binding/
> core-js-pure@3.1.3 postinstall /Users/vantino/Projects/acts-blog/node_modules/core-js-p
> node scripts/postinstall || echo "ignore"

Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaS

The project needs your help! Please consider supporting of core-js on Open Collective or
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)

> webpack-cli@3.3.2 postinstall /Users/vantino/Projects/acts-blog/node_modules/webpack-cl
> node ./bin/opencollective.js
```

Рисунок 5.3 – Встановлення пакетів npm

Під час компіляції не тільки будуть збиратися результуючі файли, а й певні файли будуть скопійовані з бібліотек до папки з публічним доступом.

Asset	Size	Chunks		Chunk Names
/css/admin.css	0 bytes	1	[emitted]	/js/app
/css/app.css	171 KiB	1	[emitted]	/js/app
/js/admin.js	167 KiB	0	[emitted]	/js/admin
/js/app.js	329 KiB	1	[emitted] [big]	/js/app
dc19ba404d20fe713	162 KiB		[emitted]	
21749319396470bde	434 KiB		[emitted] [big]	
241d60582ae9369b9	162 KiB		[emitted]	
3f8b8e6982036873e	75.4 KiB		[emitted]	
ef5805a46892932ad	95.7 KiB		[emitted]	

Рисунок 5.4 – Компіляція скриптів та стилів

Отже, тепер наявний розгорнутий проект та одного користувача для адміністрування. За змовчуванням він має наступні дані для входу:

Email: admin@acts.kpi.ua

Пароль: acts2019

5.2 Приклад використання

На даний момент вже маємо розгорнуте програмне забезпечення, але не існує користувачів. Право для створення користувачів є тільки у головного адміністратора. Для цього необхідно перейти на сторінку /admin і повинно перенаправити на сторінку з авторизацією.

Після авторизації повинна відкритися сторінка з адміністративною панеллю, що наведено на Рисунку 5.5.. Зліва можна побачити меню з посиланнями на керування профілями та користувачами.

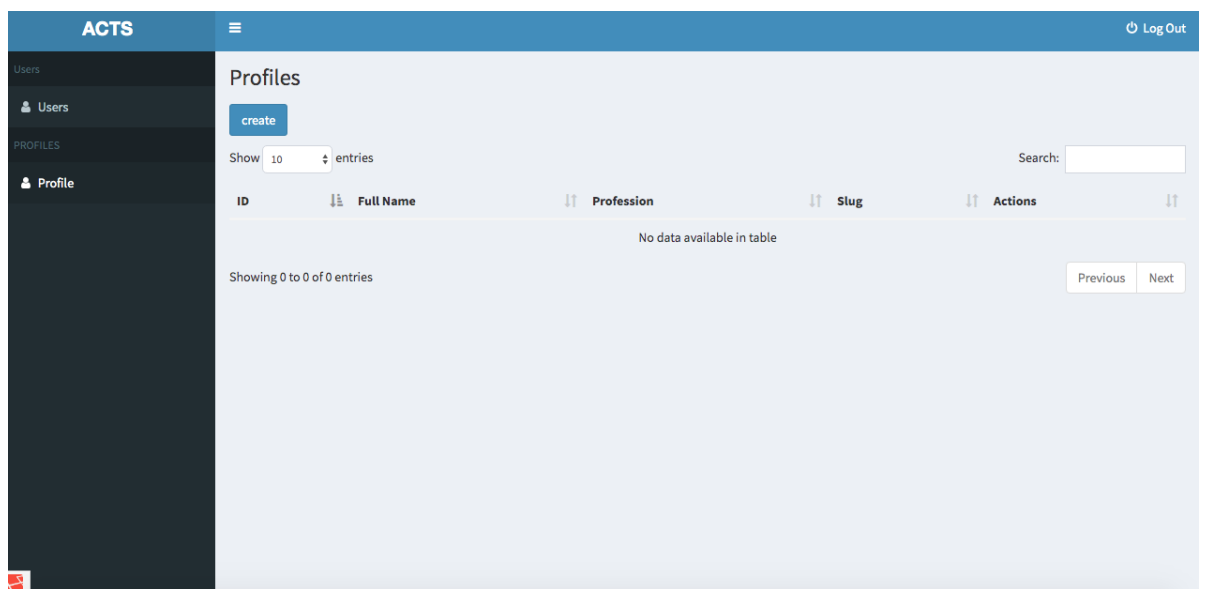


Рисунок 5.5 – Сторінка адміністратора

Далі необхідно перейти на сторінку з користувачами та натиснути на кнопку для створення. Там вводимо дані для користувача та slug який буде використовуватися у посиланні для перегляду. Форму для створення користувача наведено на Рисунку 5.6.

Рисунок 5.6 – Приклад створення користувача

Наступним кроком нам необхідно перейти на сторінку з користувачем, але оскільки ми створили лише користувача і не створили профіль, то при переході на сторінку для перегляду профіля ми отримаємо сторінку з помилкою 404.

Зараз авторизовані під користувачем адміністратора. Для того, щоб редагувати профіль нам необхідно перейти як користувачу, чий профіль плануємо редагувати. Для цього ми повинні вийти зі свого профіля за допомогою кнопки у правому верхньому кутку. Далі ми переходимо на сторінку для керування профілем, а ця сторінка має маршрут `/ {slug}/admin`.

Необхідно знову авторизуватись, але вже під новим користувачем.

Рисунок 5.6 – Сторінка створення та редагування профіля

Теперь вводимо всі необхідні данні, в форму як на Рисунку 5.6. Слід також зазначити, що ви можете прикріплювати файли та потім створювати посилання для доступу до них. Це стало можливо за допомогою сторонніх бібліотек, що у фоновому режимі завантажують файл на сервер і повертають лише посилання на цей файл. Майже таким самим чином можливо завантажити і аватарку користувача. Тоді посилання автоматично підставляється у форму для аватара.

Користуючись кнопками, можливо додавати місця роботи та навчання. Якщо значення закінчення не буде введено, то буде вважатись, що дійсне по наш час.

Після збереження, можливо перейти на сторінку профіля для перевірки. При правильному виконанні всіх попередніх дій, у браузері має відкритися сторінка схожа на Рисунок 5.7.

Слід зазначити, що для редагування знов необхідно перейти на сторінку `/[slug]/admin` під необхідним користувачем. Навіть головний адміністратор не має доступу до цієї сторінки.

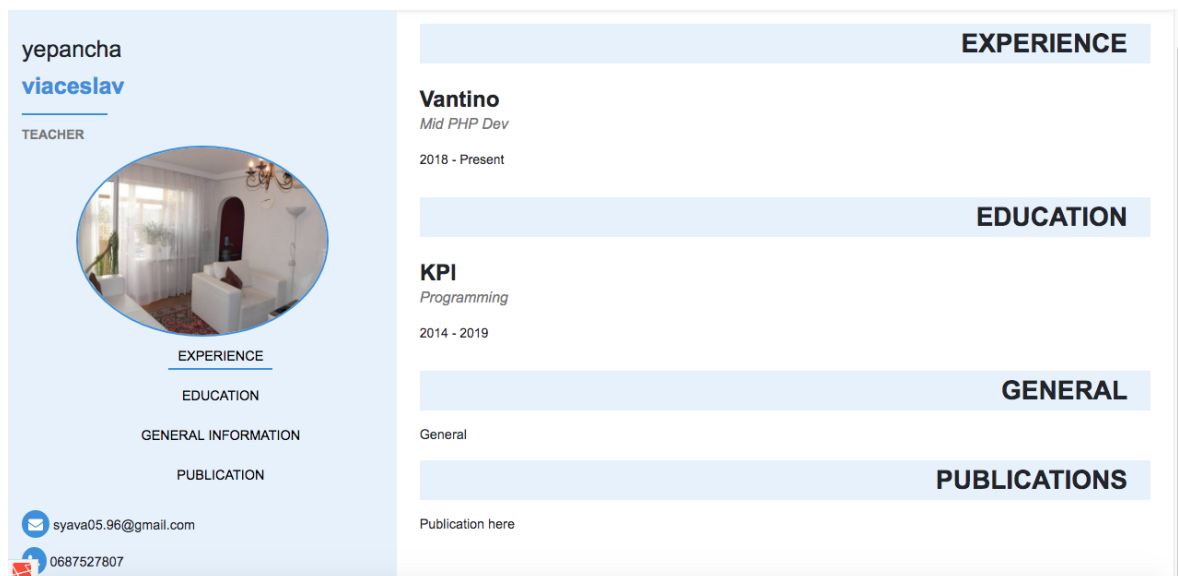


Рисунок 5.7 – Приклад створеного профіля

Отже, у цьому розділі було продемонстровано процес розгортання проекту на сервері та приклад і алгоритм його використання.

ВИСНОВКИ

В дипломному проекті була створена платформа для автоматизації учбового процесу на прикладі модуля для викладачів. Було розглянуто систему освіти в Україні, виділено основні частини, які ми можемо автоматизувати на данному етапі.

У данній роботі ми виділили один з модулів для автоматизації, що полягав у створенні мікроблогу для викладачів. З вомог до цього модулю, ми виділили, що він має бути простий, швидкий та зрозумілий.

Було глибоко проаналізовано існуючі рішення, що дозволяли швидко розробити такий мікроблог, але кожне з них включало в себе надлишковий функціонал. Спираючись на цей фактор, було винішено розробити свою систему для мікроблогу.

Сам мікроблог був реалізован мовою програмування PHP та використовуючи фреймворк Laravel. Для баз даних використовувалася СУБД PostgreSQL.

Також був описаний процес розробки, процес розгортання на сервері та приклад використання розробленого мікроблогу.

					IT41.100БАК.005 ПЗ	Лист
						59
Зм	Арк	№ докум.	Підпис	Дата		

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Енциклопедія освіти, Кремень В.Г. 2008. — 1040 с.
2. Проблеми загальної та педагогічної психології Костюк Г.С. 2010. — Т. XII, ч. 5. — С. 337–341.
3. WordPress [Електронний ресурс] : Режим доступу: <https://uk.wikipedia.org/wiki/WordPress> — Назва з екрану.
4. Joomla! [Електронний ресурс] : Режим доступу: <https://uk.wikipedia.org/wiki/Joomla!> — Назва з екрану.
5. Drupal [Електронний ресурс] : Режим доступу: <https://uk.wikipedia.org/wiki/Drupal> — Назва з екрану.
6. PHP [Електронний ресурс] : Режим доступу: <https://php.net/manual/ru/intro-what-is.php> — Назва з екрану.
7. Symfony framework [Електронний ресурс] : Режим доступу: <https://symfony.com> — Назва з екрану.
8. Laravel [Електронний ресурс] : Режим доступу: <https://laravel.com> — Назва з екрану.
9. Yii [Електронний ресурс] : Режим доступу: <https://yiiframework.com> — Назва з екрану.
10. Composer [Електронний ресурс] : Режим доступу: <https://getcomposer.org> — Назва з екрану.
11. Основи баз даних, Кузнєцов С.Д. 2001. – 14с.
12. Комп'ютерні мережі, А. Г. Микитишин, М. М. Митник, П. Д. Стухляк, В. В., 2013. — 256 с.
13. Комп'ютерні мережі, Буров Є. В., 2010. — 262 с.
14. Нейросетевые системы управления, Терехов В.А., Ефимов Д.В., Тюкин И.Ю. 2002. - 183с.
15. Composer -- Introducing [Електронний ресурс] : Режим доступу: <https://getcomposer.org/introducing> — Назва з екрану.